国外电子与通信教材系列

视频处理与通信

Video Processing and Communications

Yao Wang
[美] Jörn Ostermann 著
Ya-Qin Zhang

侯正信 杨 喜 王文全 等译





電子工業出版社
Publishing House of Electronics Industry
http://www.phei.com.cn

视频处理与通信

Video Processing and Communications

本书介绍了视频信号分析和数字视频处理的基本原理，结合国际最新科技进展对视频编码技术、国际编码标准、视频通信及网络流媒体技术进行了重点讨论，写作风格注重扎实的理论基础和复杂的系统概念之间的平衡，是目前数字视频领域的一本最具系统性、先进性和实用性的教程之一。每章末尾附有小结和习题，附录给出了有关的数学方法和部分习题答案。

作者简介

Yao Wang：美国布鲁克林理工大学电子工程系教授。现从事视频处理和传输、多媒体信号处理、医学图像处理等相关方面的研究。1990年于加州大学圣芭芭拉分校获博士学位。1998年在美国普林斯顿大学和德国Erlangen大学任访问教授。1992～2000年在AT&T实验室任兼职研究员。2000年获纽约市长优秀青年科学技术成果奖。

Jörn Ostermann：1994年在德国汉诺威大学获博士学位。1996年至今在AT&T图像处理与技术研究室从事视频编码、形状编码、多模式人机对话、标准化以及图像分析方面的研究工作。他曾任IEEE多媒体信号处理技术委员会会员，IEEE CAS（电路与系统）视觉信号处理和通信技术委员会主席。1998年获得AT&T标准认可奖和ISO（国际标准化组织）奖。

Ya-Qin Zhang：现为微软中国研究院院长。从事MPEG2/DTV、MPEG4/VLBR以及多媒体信息技术的研究与商品化工作。在国际会议和期刊上发表过200多篇论文，在数字视频、Internet、多媒体、无线和卫星通信方面获得40多项美国专利。1989年于美国George Washington 大学获博士学位。他曾任IEEE Transactions on Circuits and Systems for Video Technology总编，是IEEE会士。1998年获得美国电子工程师荣誉学会授予的“杰出青年电子工程师奖”











责任编辑：赵宏英
封面设计：毛惠庚

ISBN 7-5053-7635-7/TN・1596　定价：49.00元

国外电子与通信教材系列

视频处理与通信

Video Processing and Communications

Yao Wang

［美］ Jörn Ostermann 著

Ya-Qin Zhang

侯正信 杨喜 王文全 等译

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

内 容 简 介

本书深入系统地阐述了视频基础理论，广泛精辟地介绍了各种实际的数字视频处理和通信系统，全面地反映了国际上数字视频技术的最新进展与成果。在概述了视频技术、视频信号分析和人类视觉系统之后，论述了数字视频信号的形成和格式转换、视频信号的数学模型、摄像机和目标的运动估计、视频编码原理和各种编码方法以及多视视频等关键课题，讨论了各种视频编码标准、视频通信中的差错控制，因特网和无线网络中的流视频等视频通信的原理和技术，每章后附有小结、习题和参考文献。

本书语言流畅、图文并茂，具有系统性、先进性和实用性特点。本书可作为理工类大专院校相关专业的研究生视频处理和通信课程的教材，也可供有关科技工作者参考或自学使用。

版权贸易合同登记号：图字：01–2002–1396

图书在版编目（CIP）数据

视频处理与通信 /（美）王瑶（Yao, W.）等著；侯正信等译. –北京：电子工业出版社，2003.6

（国外电子与通信教材系列）

书名原文：Video Processing and Communications

ISBN 7–5053–7635–7

Ⅰ.视… Ⅱ.①王… ②侯… Ⅲ.视频信号–信号处理–高等学校–教材， Ⅳ. TN941.1

中国版本图书馆 CIP 数据核字（2003）第 045173 号

责任编辑：赵宏英

印 刷 者：北京牛山世兴印刷厂

出版发行：电子工业出版社 http://www.phei.com.cn

北京市海淀区万寿路 173 信箱 邮编：100036

经 销：各地新华书店

开 本：787 × 1092 1/16 印张：29 字数：742 千字

版 次：2003 年 6 月第 1 版 2003 年 6 月第 1 次印刷

定 价：49.00 元

凡购买电子工业出版社的图书，如有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系。联系电话：（010）68279077

序

2001年7月间，电子工业出版社的领导同志邀请各高校十几位通信领域方面的老师，商量引进国外教材问题。与会同志对出版社提出的计划十分赞同，大家认为，这对我国通信事业、特别是对高等院校通信学科的教学工作会很有好处。

教材建设是高校教学建设的主要内容之一。编写、出版一本好的教材，意味着开设了一门好的课程，甚至可能预示着一个崭新学科的诞生。20世纪40年代MIT林肯实验室出版的一套28本雷达丛书，对近代电子学科、特别是对雷达技术的推动作用，就是一个很好的例子。

我国领导部门对教材建设一直非常重视。20世纪80年代，在原教委教材编审委员会的领导下，汇集了高等院校几百位富有教学经验的专家，编写、出版了一大批教材；很多院校还根据学校的特点和需要，陆续编写了大量的讲义和参考书。这些教材对高校的教学工作发挥了极好的作用。近年来，随着教学改革不断深入和科学技术的飞速进步，有的教材内容已比较陈旧、落后，难以适应教学的要求，特别是在电子学和通信技术发展神速、可以讲是日新月异的今天，如何适应这种情况，更是一个必须认真考虑的问题。解决这个问题，除了依靠高校的老师和专家撰写新的符合要求的教科书外，引进和出版一些国外优秀电子与通信教材，尤其是有选择地引进一批英文原版教材，是会有好处的。

一年多来，电子工业出版社为此做了很多工作。他们成立了一个“国外电子与通信教材系列”项目组，选派了富有经验的业务骨干负责有关工作，收集了230余种通信教材和参考书的详细资料，调来了100余种原版教材样书，依靠由20余位专家组成的出版委员会，从中精选了40多种，内容丰富，覆盖了电路理论与应用、信号与系统、数字信号处理、微电子、通信系统、电磁场与微波等方面，既可作为通信专业本科生和研究生的教学用书，也可作为有关专业人员的参考材料。此外，这批教材，有的翻译为中文，还有部分教材直接影印出版，以供教师用英语直接授课。希望这些教材的引进和出版对高校通信教学和教材改革能起一定作用。

在这里，我还要感谢参加工作的各位教授、专家、老师与参加翻译、编辑和出版的同志们。各位专家认真负责、严谨细致、不辞辛劳、不怕琐碎和精益求精的态度，充分体现了中国教育工作者和出版工作者的良好美德。

随着我国经济建设的发展和科学技术的不断进步，对高校教学工作会不断提出新的要求和希望。我想，无论如何，要做好引进国外教材的工作，一定要联系我国的实际。教材和学术专著不同，既要注意科学性、学术性，也要重视可读性，要深入浅出，便于读者自学；引进的教材要适应高校教学改革的需要，针对目前一些教材内容较为陈旧的问题，有目的地引进一些先进的和正在发展中的交叉学科的参考书；要与国内出版的教材相配套，安排好出版英文原版教材和翻译教材的比例。我们努力使这套教材能尽量满足上述要求，希望它们能放在学生们的课桌上，发挥一定的作用。

最后，预祝“国外电子与通信教材系列”项目取得成功，为我国电子与通信教学和通信产业的发展培土施肥。也恳切希望读者能对这些书籍的不足之处、特别是翻译中存在的问题，提出意见和建议，以便再版时更正。

吴佑寿

中国工程院院士、清华大学教授

“国外电子与通信教材系列”出版委员会主任

出 版 说 明

进入21世纪以来，我国信息产业在生产和科研方面都大大加快了发展速度，并已成为国民经济发展的支柱产业之一。但是，与世界上其他信息产业发达的国家相比，我国在技术开发、教育培训等方面都还存在着较大的差距。特别是在加入WTO后的今天，我国信息产业面临着国外竞争对手的严峻挑战。

作为我国信息产业的专业科技出版社，我们始终关注着全球电子信息技术的发展方向，始终把引进国外优秀电子与通信信息技术教材和专业书籍放在我们工作的重要位置上。在2000年至2001年间，我社先后从世界著名出版公司引进出版了40余种教材，形成了一套"国外计算机科学教材系列"，在全国高校以及科研部门中受到了欢迎和好评，得到了计算机领域的广大教师与科研工作者的充分肯定。

引进和出版一些国外优秀电子与通信教材，尤其是有选择地引进一批英文原版教材，将有助于我国信息产业培养具有国际竞争能力的技术人才，也将有助于我国国内在电子与通信教学工作中掌握和跟踪国际发展水平。根据国内信息产业的现状、教育部《关于"十五"期间普通高等教育教材建设与改革的意见》的指示精神以及高等院校老师们反映的各种意见，我们决定引进"国外电子与通信教材系列"，并随后开展了大量准备工作。此次引进的国外电子与通信教材均来自国际著名出版商，其中影印教材约占一半。教材内容涉及的学科方向包括电路理论与应用、信号与系统、数字信号处理、微电子、通信系统、电磁场与微波等，其中既有本科专业课程教材，也有研究生课程教材，以适应不同院系、不同专业、不同层次的师生对教材的需求，广大师生可自由选择和自由组合使用。我们还将与国外出版商一起，陆续推出一些教材的教学支持资料，为授课教师提供帮助。

此外，"国外电子与通信教材系列"的引进和出版工作得到了教育部高等教育司的大力支持和帮助，其中的部分引进教材已通过"教育部高等学校电子信息科学与工程类专业教学指导委员会"的审核，并得到教育部高等教育司的批准，纳入了"教育部高等教育司推荐——国外优秀信息科学与技术系列教学用书"。

为做好该系列教材的翻译工作，我们聘请了清华大学、北京大学、北京邮电大学、东南大学、西安交通大学、天津大学、西安电子科技大学、电子科技大学等著名高校的教授和骨干教师参与教材的翻译和审校工作。许多教授在国内电子与通信专业领域享有较高的声望，具有丰富的教学经验，他们的渊博学识从根本上保证了教材的翻译质量和专业学术方面的严格与准确。我们在此对他们的辛勤工作与贡献表示衷心的感谢。此外，对于编辑的选择，我们达到了专业对口；对于从英文原书中发现的错误，我们通过与作者联络、从网上下载勘误表等方式，逐一进行了修订；同时，我们对审校、排版、印制质量进行了严格把关。

今后，我们将进一步加强同各高校教师的密切关系，努力引进更多的国外优秀教材和教学参考书，为我国电子与通信教材达到世界先进水平而努力。由于我们对国内外电子与通信教育的发展仍存在一些认识上的不足，在选题、翻译、出版等方面的工作中还有许多需要改进的地方，恳请广大师生和读者提出批评及建议。

电子工业出版社

教材出版委员会

译 者 序

近年来随着多媒体和网络通信技术的飞速发展，视频、图像、计算机视觉、多媒体数据库和计算机网络技术日益融合，遍及国民经济和社会生活的各个方面。处于其核心地位的视频处理、视频编码和视频通信，已成为信息与通信工程学科的前沿领域和热门课题。在这一领域中，新概念、新算法、新标准、新协议、新技术、新理论正在不断涌现、方兴未艾。视频处理和通信的广泛应用和深入研究，呼唤从实践到理论的升华，呼唤对新概念的确切理解和学术认同。另一方面，面对数字视频的大千世界，研究生和科技工作者迫切希望能够有一本系统讲授视频处理与通信的理论和技术的教科书作为入门的向导，跟上国际学术界的发展步伐，开展科学研究和工程实践。本书正是这样一本理想的、最新的和不可多得的教材。

本书全面地反映了国际上数字视频领域的最新进展与成果，深入系统地阐述了视频基础理论，广泛精辟地介绍了各种实际的数字视频处理和通信系统，力求在深度与广度、理论与实践之间取得平衡。在概述了视频技术、视频信号分析和人类视觉系统之后，论述了数字视频信号的形成和格式转换、视频信号的数学模型、摄像机和目标的运动估计、视频编码原理和各种编码方法以及多视视频等关键课题，讨论了各种视频编码标准、视频通信中的差错控制，因特网和无线网络中的流视频等视频通信的原理和技术。

本书是作者通过科研和教学实践编写而成的，是研究生视频处理和通信课程的首选教材，也可供有关科技工作者参考或自学使用。本书语言流畅、图文并茂。每章后附有小结和习题，书后附有部分习题解答和英文缩写词汇表。各章的参考文献为读者进一步深入钻研提供了便利。本书的翻译、出版和广泛使用，必将对我国数字视频技术的发展产生深远的影响。

本书由侯正信负责并承担主要翻译工作，参加本书翻译工作的还有杨喜、王文全、陈平、郭华南、高岩嵩，侯亭、王淑琴、何宇清、郭旭静、郭迎春、徐妮妮、李乐、张静、丁学文、李雪梅、王建惠、肖志涛、杨爱萍和王安国等。本书涉及许多新概念和新技术，由于译者水平有限，译文中可能存在一些不妥之处，敬请专家和广大读者批评指正。

序　　言

在过去10年左右的时间里，在多媒体表示和通信领域取得了惊人的进展。首先，已经很清楚的是媒体的所有方面正在“进入数字化”：从表示到传输，从处理到检索，从演播室到家庭。其次，数字多媒体的压缩和通信算法取得了显著的进步，这使得在目前的网络中以相对低的码率传输高质量的视频成为可能。第三，超大规模集成电路技术的发展可以经济有效地实现复杂的软件。最后但并非不重要的是，由ISO/MPEG和ITU-T制定的多个国际标准为不同的厂家和视频提供者奠定了一个共同工作的基础。

同时，无线和网络技术爆炸性的发展深刻地改变了全球通信的基础结构。无线、多媒体和网络的融合将彻底改变人们进行商务活动和彼此间通信的方式。未来的计算和通信结构将具有实际上的无限带宽、全连接性、高移动性和充裕的多媒体容量。

随着多媒体的逐步渗透，视频、图形、计算机视觉、多媒体数据库和计算机网络之间的界限开始变得模糊了，使视频处理成为一个多学科的激动人心的领域。今天，视频处理处于多媒体的核心地位。在其所包含的众多技术中，视频编码及其标准化在这些发展中无疑起着关键性的作用。本书包含了数字视频处理的基本原理和技术，重点是视频编码和视频通信。它可以作为研究生视频处理课程的教材，也可以作为研究人员和工程师参考或自学使用。在内容选择上，我们力求在提供一个扎实的理论基础与给出取自实际视频系统的复杂系统实例之间达到平衡。

内容提要

第1章给出了视频技术的概述，从模拟彩色电视系统到数字视频。第2章勾画了在频域进行视频分析的解析框架，并且描述了人类视觉系统的特性。第3章到第12章集中于数字视频技术的几个关键子课题。第3章和第4章考虑如何对连续空间视频信号采样，以保证在可接受的码率下保持最大的感知信息，以及如何把视频从一种格式转化为另一种格式。第5章给出了形成视频信号的各部分模型，包括摄像机、照明光源、成像目标和场景组成；摄像机和目标的三维(3D)运动以及它们向二维(2D)图像平面的投影讨论得比较详细，因为这些模型是展开运动估计算法的基础，这些是第6章和第7章的内容。第6章重点在二维运动估计，它在现代视频编码器中是关键性的部分。这也是三维运动估计的一个必要的预处理步骤。我们给出了二维运动估计的基本原理，同时给出了基于不同二维运动表示的实际算法。第7章考虑三维运动估计，它是各种计算机视觉应用所需要的，也有助于提高视频编码的效率。

第8章到第11章全部是视频编码的课题。第8章介绍了信源编码的基本原理和技术，包括有损和无损编码的信息论极限、二进制编码方法以及标量和矢量量化。第9章集中于基于波形的方法(包括变换和预测编码)，并且介绍了基于块的混合编码结构，它是所有国际视频编码标准的核心。第10章讨论了基于内容的编码，通过利用景物内容的知识这种方法有望达到极高的压缩率。第11章给出了可分级编码方法，这种方法非常适合于视频流和广播应用，因为在这些场合，接收者经常存在网络连接和计算能力的变化。第12章介绍了立体和多视角视频处理技术，包括位差估计和这些序列的编码。

第13章到第15章包括了视频通信的系统级问题。第13章介绍了H.261,H.263,MPEG-1,MPEG-2和MPEG-4视频编码标准,比较了他们的应用场合及相对性能。这些标准集中了第8章到第11章讨论的许多编码技术,同时简单地叙述了多媒体内容描述标准MPEG-7。第14章综述了对抗视频通信系统中传输差错的技术,同时描述了不同视频应用的需求以及各种网络的特性。作为实际视频通信系统的一个例子,我们以因特网和无线网络中的视频流来结束本书。第15章讨论了对流系统主要组成部分的要求和代表性的解决方案。

对教学和自学的使用建议

作为先决条件,学生应完成信号与系统、通信、概率论以及最好一门图像处理方面的本科生课程。对于一个学期的重点在视频编码和通信的课程,我们建议包括前两章,然后是视频模型(第5章),二维运动估计(第6章),视频编码(第8章到第11章),标准(第13章),差错控制(第14章)和视频流系统(第15章)。另一方面,对于一般的视频处理课程,前9章,包括前言(第1章)、频域分析(第2章)、采样和采样率转换(第3、4章)、视频模型(第5章)、运动估计(第6章和第7章)和基本视频编码技术(第8章和第9章),以及从第10章到第13章中选择一些内容(基于内容的编码、标量编码、立体和视频编码标准)可能是合适的。无论哪种情况,如果学生已经预先完成了信源编码课程,均可跳过第8章或只进行简单的回顾。第7章(三维运动估计)、第10章(基于内容的编码)、第11章(标量编码)、第12章(立体)、第14章(差错控制)和第15章(视频流)也可用做视频提高课程,而将其他章节作为先修视频课程。在所有情况下,加*号的节都可以跳过或留做优秀的学生进一步研究。

在第1章到第14章的末尾给出了习题供自学或课后作业使用。附录D给出了部分习题答案。本书的网站(www.prenhall.com/wang)可提供用来生成本书的一些附图的MATLAB原稿。教师可以修改这些原稿以生成类似的例子。这些原稿也帮助学生了解基础的操作。可以从该网站上下载视频序列样本,以使学生可以通过实际序列对各种算法的性能进行评价。网站上也包含了一些用标准算法压缩的序列,便于教师演示不同码率下不同技术的编码效应。

致谢

非常感谢帮助本书得以出版的所有同仁们。在视频编码标准方面有着丰富经验的AT&T实验室的Barry G. Haskell博士审阅了第13章,为该章以及其他章节提供了宝贵的资料。作为无线通信领域的首席专家,Polytechnic大学的David J. Goodman教授对14.2.2节提供了宝贵的资料,其中一部分是无线网络特性的总结。Southern California大学的Antonio Ortega教授和Mitsubishi电子研究所的Anthony Vetro博士以及Polytechnic大学的一位博士生建议了应该在码率控制一章中包括哪些内容,并且审阅了9.3.3节和9.3.4节。Carnegie Mellon大学的博士生Dapeng Wu先生和Fijitsu实验室的Yiwei Hou博士帮助编写了第15章。Nokia研究中心的Ru-Shang Wang博士,Mit-subishi电子研究实验室同时也是Polytechnic大学博士生的Fatih Porikli先生,以及Carnegie Mellon大学的学生Khalid Goudeaux生成了一些与立体有关的图像。Polytechnic大学的Haidi Gu先生提供了可分级视频编码的样本图像。Dorota Ostermann女士为本书的封面提供了出色的设计。

我们也感谢那些为充实本书提供了宝贵的意见和建议的未提到姓名的审阅者。同时感谢 Polytechnic 大学的学生们，他们使用了本书的稿本并指出了许多印刷错误和不一致之处。附录 D 中的答案也是基于他们的家庭作业给出的。最后，我们感谢 Prentice Hall 出版公司的 Tom Robbins 的鼓励和指导。Yao Wang 感谢国家科学基金会和纽约州通信高级技术研究中心在过去 10 年内的科研拨款，这些资助产生了包括在本书中的一些研究成果。

最重要的，我们要深深地感激我们的家人，是他们允许并鼓励我们完成了这项工作。这项工作从四年前开始，占去了我们本应和家人共度的许多珍贵时光。我们的孩子 Yana 和 Brandon 的出生使本书的出版推迟了一些时间，但也提供了完成它的动力。谨以本书献给我们的家人，感谢他们的爱、热情和支持。

YAO WANG

Polytechnic 大学，Brooklyn，纽约，美国

yao@vision.poly.edu

JöRN OSTERMANN

AT&T 实验室，Middletown，新泽西州，美国

osterman@research.att.com

YA - QIN ZHANG

微软研究中心，北京，中国

yzhang@microsoft.com

符号一览表

数学符号	
$[\mathbf{A}]$	带有方括号的粗体大写罗马字母代表矩阵
$\mathbf{a}$, $\mathbf{A}$	粗体小写或大写罗马字母代表矢量
$[\mathbf{A}]^T$, $\mathbf{a}^T$	矩阵或矢量的转置
$[\mathbf{A}]^{-1}$	矩阵$[\mathbf{A}]$的逆
$\det[\mathbf{A}]$	矩阵$[\mathbf{A}]$的行列式
$\|\mathbf{a}\|$	矢量$\mathbf{a}$的l_2范数
$\mathcal{R}^K$	K维实数空间
$\mathcal{C}^K$	K维复数空间
$\mathcal{Z}^K$	K维整数空间
$\mathcal{F}$	手写大写字母代表随机变量
$\boldsymbol{\mathcal{F}}$	手写粗体大写字母代表随机矢量$\boldsymbol{\mathcal{F}}=[\mathcal{F}_1,\mathcal{F}_2,\cdots,\mathcal{F}_N]^T$或随机序列$\boldsymbol{\mathcal{F}}=\{\mathcal{F}_1,\mathcal{F}_2,\cdots,\}$
$E\{\cdot\}$	数学期望运算
$\forall$	对于所有
Λ	点阵
Λ^*	Λ的倒数点阵
$d(\Lambda)$	Λ的密度
$[\mathbf{V}]$	为Λ生成的矩阵
$[\mathbf{U}]$	为Λ^*生成的矩阵,$[\mathbf{U}]=([\mathbf{V}]^T)^{-1}$
$\mathcal{V}(\Lambda)$	Λ的沃罗纳单元
视频表示	
$\mathbf{X}=[X,Y,Z]^T$	三维空间一点的坐标:X,Y,Z分别代表水平、垂直和深度位置
$\mathbf{x}=[x,y]^T$	二维图像平面内一个像素的坐标:x和y分别代表水平和垂直位置
t	时间指数,连续的或离散的
$\mathbf{m}=[m,n]^T$	二维数字图像中一个像素的离散坐标:m和n分别代表像素的列和行的指数
k	离散时间指数(即帧数)
$\psi(x,y,t)$	视频序列在位置$\mathbf{x}$和帧时t处的图像值。这个图像值可以是代表亮度的标量或是代表三色成分的矢量。这个符号用来表示在连续或离散空间定义的一般的视频信号
$\psi_1(x,y)$, $\psi_2(x,y)$	代表二维运动估计中的锚帧和目标帧
$\psi_p(x,y,t)$	视频编码中帧$\psi(x,y,t)$的预测图像
$\psi_r(x,y)$	用来预测某一帧的参考帧
$\psi_v(x,y,t)$	多视角系统中视角v的图像函数
$\psi(m,n,k)$	数字视频中像素位置为(m,n)、帧时为k处的图像值

(续表)

视频表示	
$\nabla\psi$	$\psi(x,y,t)$ 的空间梯度，$\nabla\psi=\left[\frac{\partial\psi}{\partial x},\frac{\partial\psi}{\partial y}\right]^T$
$\Delta_x,\Delta_y,\Delta_t$	分别表示在水平、垂直和时间方向上的采样间隔
$f_{s,t}$	时间方向上的采样频率或帧率，$f_{s,t}=1/\Delta_t$，单位是帧/秒(fps)或 Hz
$f_{s,x},f_{s,y}$	水平和垂直方向上的采样速率，$f_{s,x}=1/\Delta_x,f_{s,y}=1/\Delta_y$，单位通常是像素/图像宽和像素/图像高
R	比特率，对于视频序列单位是比特/秒(bps)，对于图像单位是比特/像素(bpp)，对于一般的离散信源是比特/采样
频域表示	
f_x,f_y	水平和垂直频率，通常单位是周/度(cpd)
f_θ	角频率，单位是 cpd
f_t	时频，单位是周/秒(cps)或 Hz
$\mathbf{f}$	多维空间的频率指数。对于视频信号，$\mathbf{f}=[f_x,f_y,f_t]^t$
$\Psi_c(f_x,f_y,f_t)$	$\psi(x,y,t)$的连续空间的傅里叶变换(CSFT)
$\Psi_s(f_x,f_y,f_t)$	$\psi(x,y,t)$的采样空间的傅里叶变换(SSFT)
$\Psi_d(f_x,f_y,f_t)$	$\psi(m,n,k)$的离散空间的傅里叶变换(DSFT)
$\Psi(f_x,f_y,f_t)$	一般用于表示视频信号的傅里叶变换，可以是 CSFT，SSFT 或 DSFT
运动/视差/摄像机特性	
$\mathbf{D}(\mathbf{X},t_1,t_2)$	点 $\mathbf{X}$ 从时间 t_1 到时间 t_2 的三维运动矢量。如果 t_1 和 t_2 已知，简写为 $\mathbf{D}(\mathbf{X})$，$\mathbf{D}$ 的 X,Y,Z 分量记为 D_x,D_y,D_z
$\mathbf{d}(\mathbf{x},t_1,t_2)$	像素 $\mathbf{x}$ 从时间 t_1 到 t_2 的二维运动矢量。如果 t_1 和 t_2 已知，简写为 $\mathbf{d}(\mathbf{x})$，$\mathbf{d}$ 的 x,y 分量记为 d_x,d_y。$\mathbf{d}(\mathbf{x};\mathbf{a})$代表运动域，它是运动参数矢量 $\mathbf{a}$ 的函数。$\mathbf{d}(\mathbf{x})$也用来表示在立体序列中两个视角之间的视差矢量
$\mathbf{w}(\mathbf{x})$	两个图像帧之间的映射函数。$\mathbf{w}(\mathbf{x})=\mathbf{x}+\mathbf{d}(\mathbf{x})$，$\mathbf{w}(\mathbf{x};\mathbf{a})$代表映射函数，它是运动参数 $\mathbf{a}$ 的函数
$\theta_x,\theta_y,\theta_z$	摄像机或目标分别相对于预定义的世界坐标 X,Y,Z 轴的旋转角度
$[\mathbf{R}]$	摄像机或目标在三维空间的旋转矩阵
$\mathbf{T}$	摄像机或目标在三维空间的平移矢量，$\mathbf{T}=[T_x,T_y,T_z]^T$
F	摄像机的焦距
$\mathbf{C}$	世界坐标下的摄像机中心

目　　录

第1章　视频的形成、感觉和表示

在第1章中，我们将叙述什么是视频信号，怎样摄取和感知它，怎样存储和传输它，以及什么是决定信号的质量和带宽（从而决定了数码率）的最重要的参数。我们首先介绍色彩感觉和再现的基础物理学（1.1节），然后叙述视频捕捉和显示的原理以及典型设备（1.2节）。大家将看到，模拟视频是以某种光栅扫描格式捕捉、存储和传输的，这种格式或者是逐行扫描或者是隔行扫描。作为例子，我们将回顾模拟彩色电视（TV）系统（1.4节），并且深入说明某些关键的参数（例如帧率和行率）是如何选择的，彩色电视信号的频谱内容是什么，信号的不同分量如何复用为一个复合信号。最后，1.5节介绍ITU-R BT.601视频格式（过去称CCIR601），它是模拟彩色电视信号的数字化形式。我们给出在选择不同的数字化参数时的一些考虑。同时也介绍几个其他的数字视频格式，包括高清晰度电视（HDTV）。本章还概述了为不同的应用所开发的压缩标准及其相应的视频格式。

本章的目的是使读者得到关于模拟和数字视频的背景知识，并且了解一般的视频系统设计问题。因此，有意采取了定性的而非定量的叙述。在后面各章中，我们将回到本章所提到的某些问题并给出较严格的表述和解释。

1.1　彩色感觉和规定

视频信号是从动态的三维景物投影到视频摄像机图像平面上的一个二维图像序列。一个视频帧中任何一点的彩色值记录了在所观察的景物中一个特定的三维点所发出或反射的光。为了理解彩色值的物理意义，本节回顾一下基础的光物理学，同时对表征光及其彩色的属性加以描述。我们也要介绍人类彩色感觉的机理，以及规定彩色信号的不同方法。

1.1.1　光和彩色

光是由人眼可以感觉的、波长在380～780纳米（nm）范围内的电磁波组成的。光的能量是用光通量测量的，单位是瓦特，它是能量发射的速率。与我们所感觉的光的亮度直接相关的是光的辐射强度，定义为以特定方向辐射到单位立体角上的光通量，度量单位是瓦特/立体角。光源通常能发射某一波长范围内的能量，并且其强度可以在时间和空间上变化。在本书中，我们使用符号$C(\mathbf{X},t,\lambda)$表示光的辐射强度分布，它指定在波长为λ、空间位置$\mathbf{X}=(X,Y,Z)$和时间t时光的亮度。

光的彩色感觉决定于其光谱成分（即它的波长组成）。例如，能量集中在700 nm附近的光显示红色，在整个可视频带内具有相同能量的光显示白色。一般来说，具有非常窄的带宽的光称为谱色，另一方面，白光被称为非彩色的。

有两种类型的光源:发射电磁波的照明光源和反射入射波的反射光源[①]。照明光源包括太阳、灯泡、电视监视器,等等。照明光源的彩色感觉取决于它所发射能量的波长范围。照明光遵循相加规则:几个混和的照明光源的彩色感觉取决于所有光源光谱的总和。例如,红光、绿光和蓝光以正确的比例组合起来形成白色光。

反射光源是那些能反射入射光(它本身也可以是反射光)的光源。当一束光照到物体上时,在某个波长范围内的能量被吸收,而剩下的则被反射出来。反射光的彩色决定于入射光的光谱成分和被吸收的波长范围。最值得注意的反射光源是彩色染料和颜料。反射光源遵循相减规则:几种混和的反射光源的彩色感觉取决于剩余的、未被吸收的波长。例如,如果入射光是白色的,那么吸收 700 nm 附近波长的染料看起来是青色的。在这种意义上,我们说青色是红色的补色(即白色减去红色)。类似地,品色和黄色分别是绿色和蓝色的补色。青色、品色和黄色染料混和形成黑色,它吸收整个可视光谱。

1.1.2 人类的彩色感觉

人类对光的感觉开始于位于视网膜(眼球内部的后表层)的感光细胞。有两种类型的感光细胞:锥状细胞和杆状细胞。锥状细胞在亮光下起作用,能感受彩色色调;杆状细胞工作在低环境光下,只能够提取亮度信息。来自视网膜的视觉信息通过视神经纤维传送到大脑的视觉皮层,在那里完成视觉的处理和理解。有三种类型的锥状细胞,它们在可见光谱上具有重叠的通带,其峰值分别位于红色(570 nm 附近)、绿色(535 nm 附近)和蓝色(445 nm 附近)波长,如图 1.1 所示。这些感光细胞对入射光分布的响应可用如下公式描述:

$$C_i = \int C(\lambda) a_i(\lambda) d\lambda, i = r, g, b \tag{1.1.1}$$

这里,$a_r(\lambda)$,$a_g(\lambda)$,$a_b(\lambda)$是指红色、绿色和蓝色锥状细胞的频率响应或相对吸收函数。这三种类型的感光细胞的组合可使人类感知任何彩色。这意味着被感知的彩色只依赖于 C_r,C_g,C_b 这三个数,而不是整个的光谱。这就是所谓彩色视觉的三感光细胞原理,是由 Young 首先提出的[17]。

描述人类彩色感觉的属性有两个:亮度和色度。亮度指被感知的光的明亮度,它是与可视频带中的总能量成正比的。色度描述被感知的光的颜色和深浅,它是由光的波长成分决定的。色度进一步由两个属性表征:色调和饱和度。色调指彩色的颜色,它是由光的峰值波长决定的,而饱和度指的是颜色有多纯,它是由光谱的范围或带宽决定的。在本书中,我们用"彩色"这个词指光的亮度和色度这两个属性,尽管通常彩色这个词只是指光的色度。

实验表明在人类视觉系统(HVS)中存在第二个处理阶段,它把锥状细胞获得的三个彩色值转换成一个与亮度成正比的值和另外两个响应于色度感觉的值,这就是所谓的 HVS 反彩色模型[2,8]。已经发现同样大小的能量在不同的波长会产生不同的亮度感觉,这种依赖于波长的亮度感觉的变化用相对亮度效率函数 $a_Y(\lambda)$表征,也如图 1.1 所示。它本质上是所有三种类型锥状细胞频率响应的和。我们可以看到绿色波长对亮度感觉的贡献最大,其次是红色波长,而蓝色波长最小。亮度(通常用 Y 表示)与入射光谱的关系是:

---

① 照明和反射光源也分别称为原发和继发光源。我们不使用这些术语是为了避免与光有关的原色相混淆。在其他资料中,照明和反射光分别被称为加色和减色。

$$Y = \int C(\lambda) a_y(\lambda) d\lambda \tag{1.1.2}$$

在上述公式中，我们忽略了时间和空间的变化，因为我们只关心在固定的空间和时间位置上的彩色和亮度感觉。我们也略去了通常与每个等式联系的比例因子，它取决于描述彩色强度和亮度的单位。

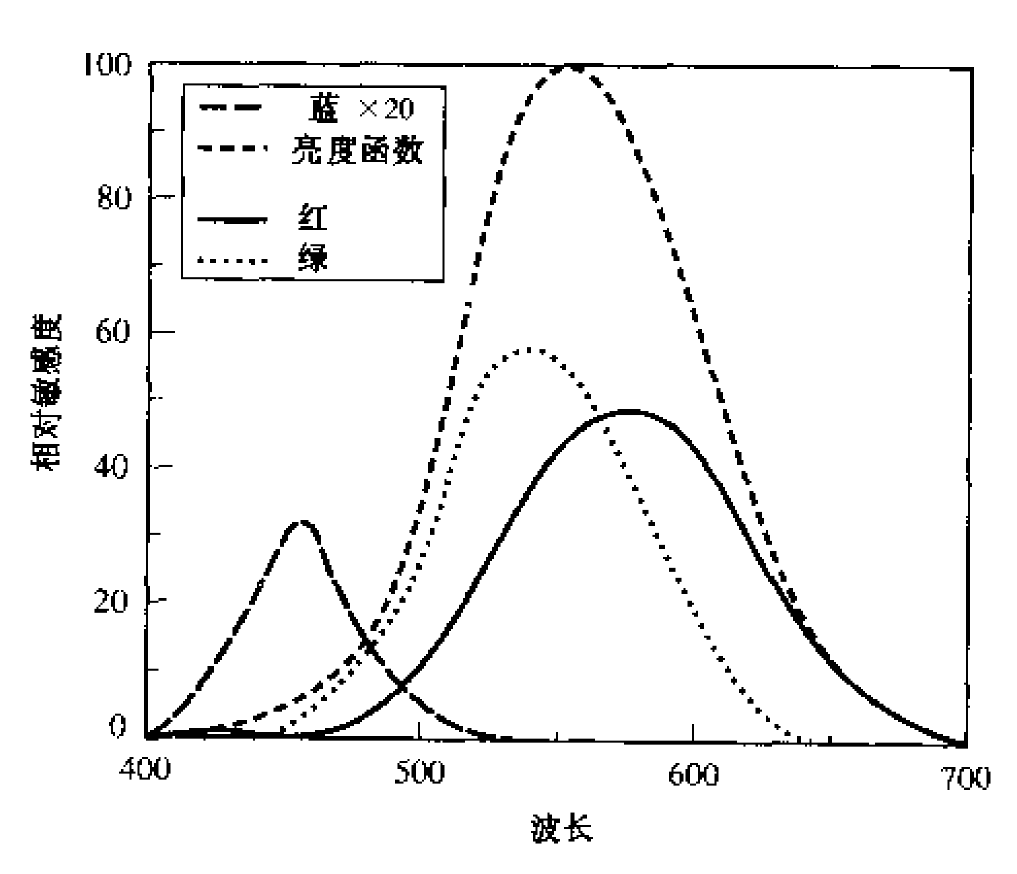


图1.1 人类视网膜中三类锥状细胞的频率响应和亮度效率函数。图中蓝色响应曲线被放大了20倍

1.1.3 彩色混合的三基色原理

彩色物理学中一个非常重要的发现是大多数彩色可由适当选择的三种基色混和产生。这就是所谓彩色混和的三基色原理，首先由 Maxwell 在 1855 年[8,16]给出证明。令 $C_k, k = 1,2,3$ 代表三种基色源的彩色，C 是一种给定的彩色。那么该原理实质上是说：

$$C = \sum_{k=1,2,3} T_k C_k \tag{1.1.3}$$

这里 T_k 是配出彩色 C 所需的三种基色的量值，称为三色激励值。一般来说，某些 T_k 可以是负的。假定只有 T_1 是负的，这意味着通过混合 C_1, C_2, C_3 不能配出 C，但是可以用 $T_2 C_2 + T_3 C_3$ 配出 $C + |T_1| C_1$。实际上应该这样来选择基色，即能够通过基色正的组合产生大多数的自然彩色。照明光源最流行的基色系包括红色、绿色和蓝色，称为 RGB 基色。反射光源最通常的基色系包括青色、品色和黄色，称为 CMY 基色。事实上，RGB 和 CMY 基色系是互补的，也就是说混合一个色系中的两种彩色会产生另外一个色系中的一种彩色。例如，红色和绿色混合会产生黄色。这种互补信息可以用一个色轮给出最好的图示，这个图可在许多图像处理书籍中找到，例如参考文献[8,3]。

对于一个选定的基色系，一种确定任何彩色三色激励值的办法是首先为基色 $C_i, i = 1,2,3$ 确定彩色匹配函数 $m_i(\lambda)$。这些函数对于整个可视频带内的各个波长 λ 描述了其谱色的三色激励值，并且可在受控观测条件下通过视觉实验确定。然后，具有光谱 $C(\lambda)$ 的任何彩色的三色激励值可由下式得到[8]：

$$T_i = \int C(\lambda)m_i(\lambda)d\lambda, i = 1,2,3 \quad (1.1.4)$$

为了通过正混合产生所有可视彩色,与基色相关的匹配函数必须是正的。

上述原理构成了彩色摄取和显示的基础。为了记录入射光的彩色,摄像机必须有三个传感器,其频率响应类似于所选基色系的彩色匹配函数。这可由具有期望频率响应的光学或电子滤波器来实现。类似地,为了显示一幅彩色图像,显示设备必须发出三束具有适当强度的所选基色的光束。实际上,是用电子束激发具有红色、绿色和蓝色的荧光粉。目前所有的显示系统都采用 RGB 基色,尽管为基色规定的标准光谱可能会有少许不同。同样地,彩色打印机可以通过以适当比例混合具有所选基色的三种颜料来产生不同的彩色。多数打印机采用 CMY 基色。为了更鲜艳和宽范围的彩色再现,一些彩色打印机在 CMY 基色中增加了黑色(K),从而采用 4 种基色。这就是所谓的 CMYK 基色,它能更真实地再现黑色。

1.1.4 由三色激励值规定彩色

三色激励值 彩色的激励表示规定了为产生期望的彩色所需要的三基色的比例系数,即公式 1.1.3 中的 T_k。为了使彩色的规定独立于基色的绝对能量,这些值应该归一化,使得当 $T_k = 1, k = 1,2,3$ 时对应具有单位能量的基准白色(所有波长的能量相同)。当我们采用 RGB 基色时,三色激励值通常记为 R, G 和 B。

彩色值 上述激励表示混合了彩色的亮度和色度属性。为了只测量光的色度信息(色调和饱和度),彩色坐标定义为:

$$t_k = \frac{T_k}{T_1 + T_2 + T_3}, k = 1,2,3 \quad (1.1.5)$$

由于 $t_1 + t_2 + t_3 = 1$,所以两个彩色值就足以规定彩色的色度。

显而易见,图像点的彩色值依赖于采用的基色。为了使彩色的描述和规定标准化,已经定义了几种标准的基色系统。例如,彩色科学家国际组织 CIE[①] 定义了一个 CIE RGB 基色系统,它包含 700(R_0),546.1(G_0)和 435.8(B_0) nm 波长的彩色。

彩色坐标转换 我们能够把基于一种基色系的彩色值转换为基于另一种基色系的彩色值。例如,为了打印以(R, G, B)坐标存储的彩色图像,经常需要将(R, G, B)坐标转换到(C, M, Y)坐标。只要将一种基色系的激励表示用另一种基色系给出,我们就能够确定这两种坐标系之间的转换矩阵。彩色转换原理与两种基色系之间的转换矩阵推导可参见文献[8]。

1.1.5 由亮度和色度属性规定彩色

通常用于彩色显示的 RGB 基色混合了光的亮度和色度属性。在许多应用中,为了能够更高效地处理和传输彩色信号,希望将亮度与色度分量分离开来描述一种彩色。为了达到这个目的,人们开发了各种三分量彩色坐标,其中一个分量反映亮度,另外两个分量共同表示色度和饱和度。CIE XYZ 基色就是这样的一个坐标,其中 Y 直接度量亮度强度。这个坐标系中的(X, Y, Z)值与 CIE RGB 坐标系中的(R, G, B)值的关系为:

① CIE 是 Commission Internationale de L'Eclariage 的缩写(国际照明委员会)。

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 2.365 & -0.515 & 0.005 \\ -0.897 & 1.426 & -0.014 \\ -0.468 & 0.089 & 1.009 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1.1.6}$$

除了能分离亮度和色度信息，CIE XYZ 系统的另一个优点是几乎所有的可见彩色都能由非负的激励值规定，而这正是人们期望的特性。问题是这样定义的 X, Y, Z 彩色并不能由实际的彩色激励所实现。因此，XYZ 基色不能直接用于产生彩色，而它主要用于定义其他的基色和彩色的数字说明。正如我们将要看到的，用于传输彩色电视信号的彩色坐标，例如 YIQ 和 YUV，都是从 XYZ 坐标导出的。

也有其他的彩色表示，除亮度外直接定义色度和饱和度。HSI 坐标就是其中之一，这里"H"表示色度，"S"表示饱和度，"I"表示强度（等价于亮度）[①]。尽管这种彩色坐标清楚地分离了光的不同属性，但是与激励值是非线性关系并且很难计算。Gonzalez 和 Woods[3]详细介绍了各种彩色坐标及它们之间的转换。

1.2 视频的捕捉和显示

1.2.1 彩色视频成像原理

在定义了什么是光以及如何感知和表征它以后，我们现在可以开始理解视频信号的意义了。简言之，视频[②] 记录了从一个观测系统（人眼或摄像机）所观测的场景中的物体发射或反射的光的强度 $C(\mathbf{X}, t, \lambda)$。一般地说，该强度在时间和空间上都有变化。这里我们假定场景中有某种照明光源。否则，将没有入射光也没有反射光，整个图像将是黑的。当由摄像机观测场景时，只有摄像机敏感的那些波长是可见的。令 $a_c(\lambda)$表示摄像机的光谱吸收函数，那么摄像机可见的三维空间中的光强度分布为：

$$\bar{\Psi}(\mathbf{X}, t) = \int_0^\infty C(\mathbf{X}, t, \lambda) a_c(\lambda) d\lambda \tag{1.2.1}$$

在任何时间 t 处，被摄像机摄取的图像函数都是在三维场景中的光分布在二维图像平面上的投影。令$\mathcal{P}(\cdot)$表示摄像机的投影算子，那么三维点 $\mathbf{X}$ 投影的二维位置由 $\mathbf{x} = \mathcal{P}(\mathbf{X})$给出。进一步，令$\mathcal{P}^{-1}(\cdot)$表示逆投影算子，那么 $\mathbf{X} = \mathcal{P}^{-1}(\mathbf{x})$就规定了与二维点 $\mathbf{x}$ 相应的三维位置。这样投影图像与三维图像的关系为：

$$\psi(\mathcal{P}(\mathbf{X}), t) = \bar{\Psi}(\mathbf{X}, t) \text{ 或 } \psi(\mathbf{x}, t) = \bar{\Psi}(\mathcal{P}^{-1}(\mathbf{x}), t) \tag{1.2.2}$$

函数$\psi(\mathbf{x}, t)$称为视频信号。我们可以看到，它描述了在时间 t 投影到图像平面 $\mathbf{x}$ 的三维位置 $\mathbf{X}$ 的辐射强度。通常，视频信号具有一个有限的时间和空间范围。空间范围依赖于摄像机的观测区域，而时间范围依赖于场景被摄取的持续时间。图像平面中的一点被称为像素（pixel，即图像元素）或简单地记为 pel[③]。对于大多数摄像系统，投影算子$\mathcal{P}(\cdot)$可近似为一个透视投影（这个问题将在 5.1 节中详细讨论）。

---

① HSI 坐标也称为 HSV，这里"V"表示强度值。

② 在本书中，当我们提到"视频"而不加其他限定时，我们指的是一个被记录的运动图像序列。

③ 严格地讲，符号 pixel 或 pel 只定义在数字图像学中，在那里视频记录中的每一幅图像或每一帧由一个有限的二维像素矩阵表示。

如果摄像机吸收函数等同于人类的相对亮度效率函数，即 $a_c(\lambda)=a_r(\lambda)$，则形成亮度图像。如果吸收函数在一个窄带中是非零的，则形成单色(或单色调)图像。为了感知所有的可见彩色，根据三基色的视觉原理(见1.1.2节)，需要三个频率响应类似于所选基色的彩色匹配函数的传感器。如前所述，多数彩色摄像机利用红色、绿色和蓝色传感器来获得彩色。

如果摄像机只有一个亮度传感器，那么$\psi(\mathbf{x},t)$是表示投影光亮度的标量函数。在本书中，我们用灰度级这个术语来表示这样一幅图像，术语黑和白将严格地用于描述只有两种彩色黑和白的图像。另一方面，如果摄像机有三个分离的传感器，每一个接收一种所选的基色，那么信号就是一个在每个点包含三个彩色值的矢量函数。我们可以利用前面章节中所介绍的其他的彩色坐标(每个包含三个值)来表征光，而不是直接规定这些彩色值。

注意，有时为了特殊的目的，我们可以采用频率不在人眼可见范围内的传感器。例如在X光成像中，传感器要对X光的光谱范围很敏感。另一方面，红外线摄像机对红外区域敏感，能在很低的环境光下工作。这种摄像机能够"看到"人眼不能感觉的物体。另一个例子是测距摄像机，其传感器发出激光束并测量它到达物体和返回传感器所需的时间。因为这个来回的时间正比于传感器到物体表面的距离，所以测距图像内任何一点的图像亮度就描述了相应的三维点到摄像机的距离或射程。

1.2.2 视频摄像机

目前所有的模拟摄像机都是一帧一帧地捕捉视频，帧与帧之间有一定的时间间隔。一些摄像机(例如电视摄像机和消费类的便携式视频摄像机)通过扫描具有一定行间隔的相继的行来获得一帧。类似地，所有显示设备都以相继的一系列的帧显示视频；对于电视监视器，扫描线相继地回扫形成分离的行。这种摄取和显示机制是根据如下事实设计的，即HVS不能感知很高频率的时间和空间的变化(HVS的这个性质将在2.4节中更深入地讨论)。

有两种基本类型的摄像机：(1)基于光电管的摄像机，例如光导摄像管、氧化铅摄像管或正析摄像管；(2)固态传感器，例如电荷耦合器件(CCD)。摄像机镜头把场景中的图像聚焦到摄像机析像管的光敏表面，由它将光信号转换成电信号。析像管的光敏表面一般是用电子束或其他的电子方法一行一行地扫描(称为光栅扫描)，然后每一帧中的扫描行被转换成用不同电压代表不同光强度的电子信号。因此不同的扫描行是以相继的方式在略微不同的时间上摄取的。对于逐行扫描，电子束是相继地扫描每一行；而对于隔行扫描，电子束是在半帧(一场)的时间隔一行扫描一行，然后扫描另外一半的行(我们将在1.3节详细讨论光栅扫描。)对于CCD摄像机，光敏表面由二维传感器矩阵组成，每个传感器对应一个像素，到达每个传感器的光信号被转换成一个电信号。在每帧时间内摄取的传感器值首先存储在缓冲器中，然后一次一行地相继读出，以形成光栅信号。与基于光电管的摄像机不同，同一帧中所有读出的值是同时被摄取的。对于隔行扫描的摄像机，每场中的扫描行是被交替读出的。

为了摄取彩色，通常有三种类型的光敏表面或CCD传感器，每个传感器的频率响应决定于所选基色的彩色匹配函数，如1.1.3节所述。为了减少成本，多数消费类摄像机采用单片CCD进行彩色成像。这是通过把每个像素的传感器区域分成3个或4个子区域，每个子区域感应于不同的基色实现的。摄取的三个彩色信号可以转换成一个亮度信号和两个色度信号，并作为分量彩色图像发送出去，或者复用成一个复合信号。这方面的内容将在1.2.4节进一步解释。

现在许多摄像机是基于 CCD 的，因为它们为获得同样的空间分辨率可以比基于光电管的摄像机做得更小和更轻。CCD 技术的优点使它能够以很小的芯片摄取很高分辨率的图像矩阵。例如，消费者使用的便携式摄像机通常采用 38 万像素的 1/3 英寸的 CCD，而 200 万像素的 2/3英寸的 CCD 已经开发用于 HDTV。基于光电管的摄像机较大且较昂贵，只用于一些特殊的应用场合，例如需要很高分辨率或需要在低环境光下有高敏感度的场合。除了彩色成像电路，多数摄像机还能实现彩色坐标转换（从 RGB 转换到亮度和色度）以及复合亮度和色度信号。为进行数字输出，也包含有模拟到数字（A/D）转换。包括在专业视频摄像机中的这种典型处理如图 1.2 所示。该摄像机提供了模拟和数字两种输出模式，而模拟输出又包括分量（CN）和复合（CS）两种格式。为了提高图像质量，该摄像机引进了数字处理技术。用于三个 CCD 中每一个 CCD 的 A/D 和预处理（包括点和区域的处理，数据检测和校正）都工作在 CCD 的输出速率 $f_{s,1}$。在它后面是以 2 倍 CCD 输出速率进行的图像增强和非线性滤波。为了匹配标准摄像机的输出速率 $f_{s,2}$（对于 1.5.2 节讨论的 ITU-R BT.601 数字视频 $f_{s,2}$ = 13.5 MHz），需要 $2f_{s,1}$ 到 $f_{s,2}$ 的速率转换。关于视频摄像机和显示技术的详细说明见文献[5]。

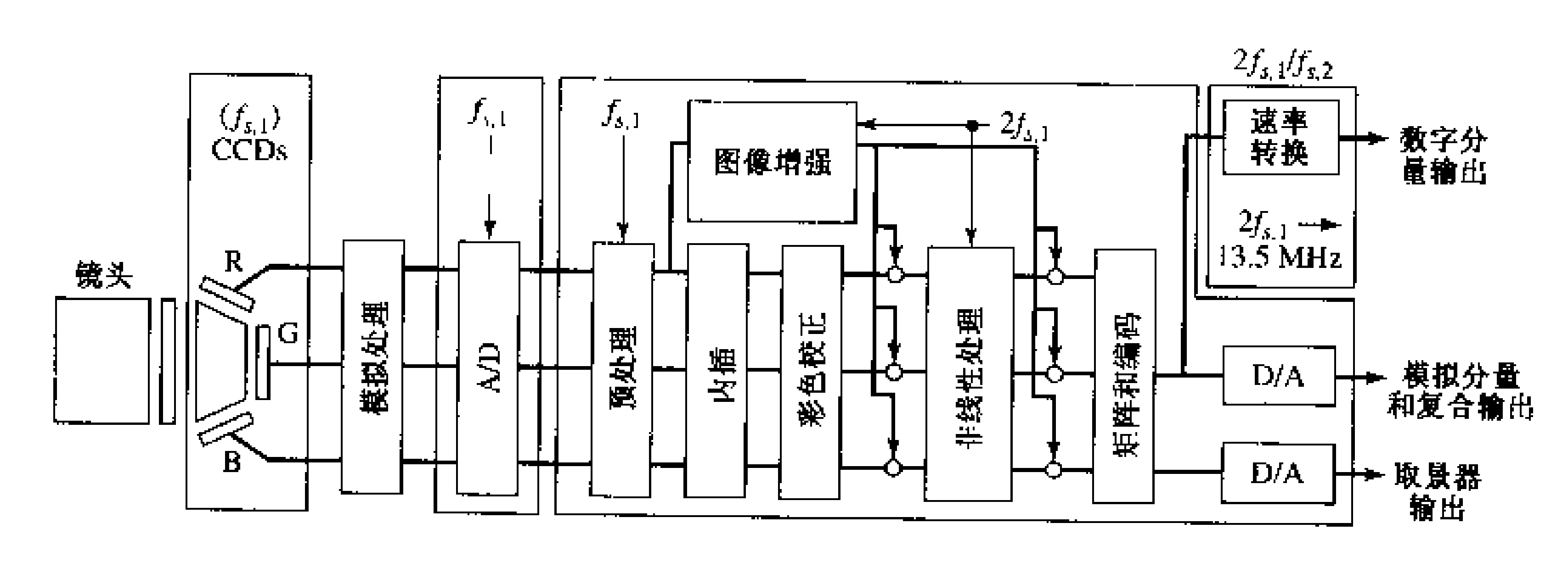


图 1.2 专业彩色视频摄像机方块示意图

1.2.3 视频显示

为显示视频，最普通的设备是阴极射线管（CRT）。在 CRT 监视器中，电子枪一行一行地向屏幕发射电子束以激励荧光粉，其强度正比于在相应位置的视频信号的强度。为了显示彩色图像，三个电子枪发射三个电子束，在每个位置以期望的强度组合激励红色、绿色和蓝色荧光粉。为了更精确，每个彩色像素由排列成小三角形的三个元素组成，称之为三元组。

CRT 能产生有很大动态范围的图像，因此显示的图像可以很亮，足以在白天或在远处观看。然而，为使电子到达屏幕的边界，CRT 的深度需要大致与屏幕的宽度相当。这样大屏幕监视器就太庞大了，不适于应用在小型轻便的设备中。为克服这个问题，人们开发了各种不同的平板显示器。一种流行的设备是液晶显示器（LCD）。LCD 的主要思想是通过施加电场改变光学特性进而改变液晶的亮度或彩色。电场由一个晶体管阵列产生或刷新，例如在 LCD 中采用驱动矩阵薄膜晶体管（TFT）或采用等离子。等离子技术消除了对薄膜晶体管的需求，从而使大屏幕 LCD 成为可能。也有平面 CRT 的新设计。关于视频显示技术更详细的描述可参见参考文献[5]。

刚刚介绍的光栅扫描和显示机制只应用于视频摄像机和显示器。对于运动图像摄像机，

摄像机在任何帧瞬间摄取的彩色图案全部记录在胶片上。显示时,用模拟光学投影系统回放相继记录的帧。

1.2.4 复合视频与分量视频

理想情况下,无论是三色激励表示还是亮度色度表示,彩色视频都应该由三个函数或信号规定,每个描述一个彩色分量。这种格式的视频称之为分量视频。主要是由于历史的原因,也存在各种复合视频格式,其中的三个彩色信号被复用成一个单独的信号。当彩色电视系统首次开发出来时发明了这种复合格式,当时要求彩色电视信号的传输方式能使黑白电视机从中抽取亮度分量。构造复合信号依赖于这样一个性质,即色度信号具有比亮度分量小得多的带宽。通过将每个色度分量调制到一个位于亮度分量高端的频率上,并把已调色度分量加到原始亮度信号,就产生了一个包含亮度和色度信息的复合信号。为了在彩色监视器上显示复合视频信号,用滤波器把已调色度信号从亮度信号中分离出来,然后把产生的亮度和色度信号转换成红色、绿色和蓝色分量。对于灰度级监示器,只提取亮度信号并直接显示。

现在所有模拟电视系统都以复合格式传输彩色电视信号。复合格式也用于将视频存储在某种模拟磁带上(例如 VHS)。除了与灰度级信号兼容以外,复合格式消除了处理彩色信号时使不同的彩色分量同步的需要。复合信号的带宽比三个分量信号带宽的总和小,因此能被更有效地传输或存储。然而这些优点是以降低图像质量为代价的:经常由于色度和亮度分量的串扰而形成可觉察的人工痕迹。

作为数码率与图像质量之间的一个折中,发明了 S-video,它由两个分量组成:亮度分量和由两个原始色度信号复合成的一个单独的色度分量。许多高级消费类视频摄像机和显示设备都能以 S-video 格式记录或显示视频。分量格式只用于专业的视频设备。

1.2.5 伽马校正

我们说过由摄像机摄取的视频帧反映了成像景物的彩色值。事实上,多数摄像机输出的信号并不是与实际的彩色值成线性关系,而是一种非线性的形式[①]:

$$v_c = B_c^{-\gamma_c} \tag{1.2.3}$$

这里 B_c 表示实际的光亮度,v_c 是摄像机的输出电压。γ_c 的取值范围从 1.0(对于多数 CCD 摄像机)到 1.7(对于光导管摄像机)[6]。类似地,大多数显示设备的输入电压值与显示的彩色强度之间也有这种非线性的关系,即:

$$B_d = v_d^{\gamma_d} \tag{1.2.4}$$

对于 CRT 显示器,典型的 γ_d 是 2.2~2.5[6]。为了呈现真实的彩色,我们必须在摄像机输出端加入一个相反的幂函数。类似地,在发送要显示的真实图像值之前,我们必须对显示设备的"伽马效应"进行预补偿。这些过程被称为伽马校正。

理想情况下的电视广播,在电视广播的发送端,被电视摄像机摄取的 RGB 值应该首先基于摄像机的伽马值进行校正,然后转换到用于传输的彩色坐标(对于 NTSC 是 YIQ,对于 PAL 和 SECAM 是 YUV)。在接收端,收到的 YIQ 或 YUV 值应该首先被转换成 RGB 值,然后用监视器

---

① 更精确的关系是 $B_c = cv_c^{\gamma_c} + B_0$,这里 c 是增益因子,而 B_0 是光强截止电平。当我们假定输出电压值经过了适当的平移和缩放时,则所给出的公式是正确的。

的伽马值补偿。然而实际上，为了在数以百万计的接收机中减少这种处理过程，广播的视频信号在 RGB 域进行预伽马校正。令 v_c 表示摄像机摄取的 RGB 信号，则用于显示的已经过伽马校正的信号由下式得到：

$$v_d' = v_c^{\gamma_c/\gamma_d} \tag{1.2.5}$$

在多数电视系统中，采用比率 $\gamma_c/\gamma_d=2.2$。这是基于假设所用的 CCD 摄像机的 $\gamma_c=1$，而 CRT 显示器的 $r_d=2.2$[6]。这些已经过伽马校正的值在传输时要转换成 YIQ 或 YUV 值。接收机在显示时只简单地用彩色坐标转换来获得 RGB 值。值得注意的是，这个过程在转换到 YIQ/YUV 域之前对显示器加了伽马校正，而这并不是严格的校正。但是这种失真并不显著，普通观众是觉察不到的[6]。

1.3　模拟视频光栅

正如我们已经说明的，现在的模拟电视系统使用光栅扫描进行视频摄取和显示。由于这是最通用的模拟视频格式，本节我们较详细地描述这种光栅扫描机制，包括逐行和隔行扫描。我们也会举例说明在不同的模拟电视系统中所用到的视频格式。

1.3.1　逐行与隔行扫描

逐行扫描　在光栅扫描中，摄像机通过时间和垂直方向上的采样摄取一个视频序列。产生的信号以一个连续的一维(1-D)波形存储。如图 1.3(a)所示，模拟视频摄像机的电子束或光束连续地扫描图像区域，从顶部到底部然后再回到顶部。得到的信号是由一系列以常规的帧间隔 Δ_t 分开的帧组成的，并且每一帧是由一系列相继的以常规的垂直间隔分开的水平扫描行组成的。每个扫描行实际上是稍微倾斜的。另外，底行比同一帧的顶行大约晚一帧的时间间隔。然而，为了分析的目的，我们通常假设一帧中所有的行是同时被采样的，并且每行是完全水平的。沿着相继帧的邻接的扫描行摄取的亮度值形一个一维模拟波形，称为光栅扫描。对于彩色摄像机，三个一维光栅转换成一个复合信号，它是一个彩色光栅。

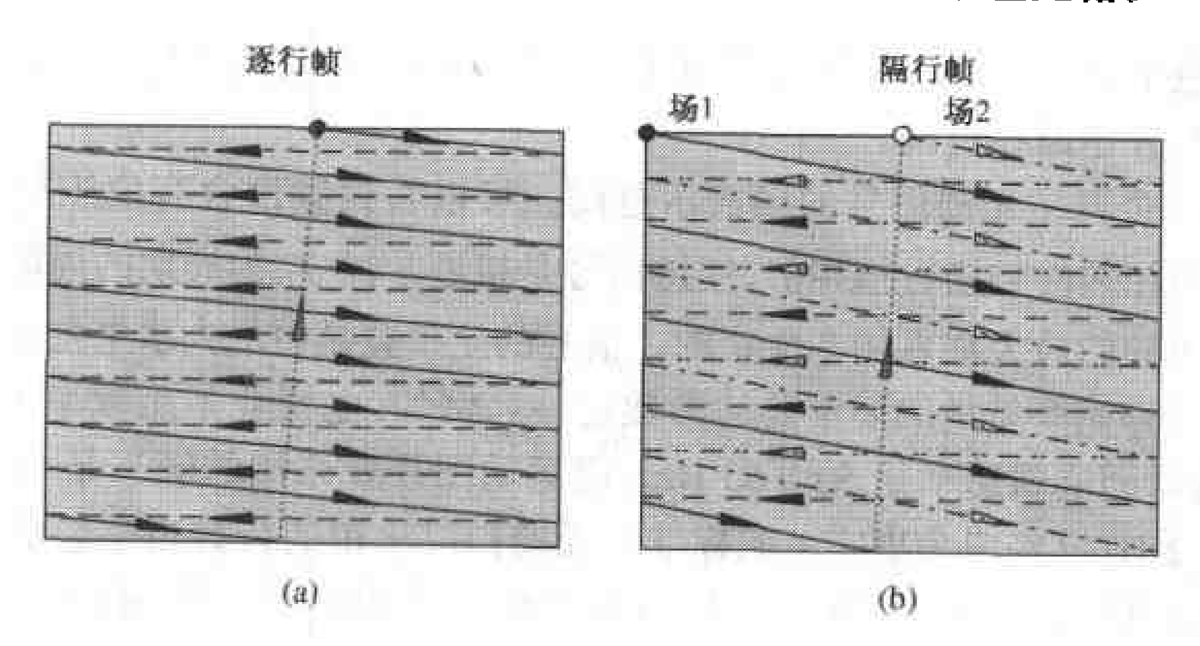


图 1.3　逐行(a)和隔行(b)光栅扫描格式

隔行扫描　上面描述的光栅扫描格式更确切地称为逐行扫描(也被称为连续或非隔行扫描)，在逐行扫描中，水平行被相继扫描。在隔行扫描中，每帧分成两场，每场包含一帧中半数

的扫描行。每两场之间的时间间隔，即场间隔，是帧间隔的一半，而在一场中的行间隔是一帧中所期望的行间隔的二倍。两个相继场的扫描行平移了每场的半个行间距，如图 1.3(b)所示。根据用于 MPEG 标准的术语，我们把一帧中包含第一行和后面各个隔行的场称为顶场，而把包含第二行和后面各个隔行的场称为底场①。在某些系统中，顶场先采样，而在另一些系统中，底场先采样。重要的是要记住，一帧中两个相邻的行在时间上是以场间隔分开的。这个事实导致在包含具有垂直边界的快速运动物体的隔行视频图像中产生难看的之字形人工痕迹。采用隔行扫描的动机是在给定时间内给定所能记录的总行数的条件下，用牺牲垂直分辨率来提高时间分辨率。在采样效率方面对隔行和逐行扫描更全面的比较将在 3.3.2 节给出。

刚才介绍的隔行扫描应该更准确地称为 2:1 隔行。通常，我们可以把一帧分成 $K \geqslant 2$ 场，每场的时间间隔是 Δ_t / K。这被称为 K:1 隔行，K 称为隔行阶数。在数字视频中，每行都是用离散采样点表示的，同一行的采样点可能会出现在不同的场中。例如，一帧中的采样点可能会以棋盘格图案分成两场。隔行阶数最广义的定义是一帧中的采样点数与一场中的采样点数的比率。

1.3.2 视频光栅特性

光栅是用两个基本参数描述的：帧率（帧/秒或 fps 或 Hz）用 $f_{s,t}$ 表示，行数（行/帧或行/像高）用 $f_{s,y}$ 表示。这两个参数定义了光栅在时间和垂直方向上的采样率。从这些参数中，我们能导出另一个重要的参数，即行率（行/秒），用 $f_l = f_{s,t} f_{s,y}$② 表示。我们也能导出时间采样间隔 $\Delta_t = 1/f_{s,t}$ 或帧间隔，垂直采样间隔或行间距 Δ_y = 像高$/f_{s,y}$，以及行时间间隔 $T_l = 1/f_l = \Delta_t / f_{s,y}$，它是扫描一行所用的时间。注意，时间间隔 T_l 包括传感器从行尾移到下一行开始处的时间，称为水平回扫时间或只称为水平回扫，用 T_h 表示。一行的实际扫描时间是 $T'_l = T_l - T_h$。同样，帧间隔 Δ_t 包括传感器从一帧中底行的末尾移到下一帧顶行的开始处的时间，称为垂直回扫时间或只称为垂直回扫，用 T_v 表示。一帧时间内实际扫描的行数称为有效行，即 $f'_{s,y} = (\Delta_t - T_v) / T_l = f_{s,y} - T_v / T_l$。通常，$T_v$ 选为 T_l 的整数倍。

隔行光栅信号的典型波形如图 1.4(a)所示。注意到信号在水平和垂直回扫期间的部分保持在相应于黑色电平之上的一个恒定的电平上。这些部分称为同步信号。显示设备在检测到这些同步信号后开始回扫过程。

图 1.4(b)是一个典型光栅信号的频谱示意图。可以看到频谱含有位于行率 f_l 及其谐波上的峰。这是因为相邻的扫描行很相似，使得信号是接近于以 T_l 为周期的周期信号。每个谐波波瓣的宽度是由帧的最大垂直频率决定的。信号的全带宽是由最大水平空间频率决定的。

帧率是决定视频光栅质量的最重要的参数之一。例如，电视业采用帧率为 25 ~ 30 Hz 的隔行扫描，有效的时间刷新速率是 50 ~ 60 Hz。而运动图片业采用的帧率是 24 Hz③。另一方面，在计算机业，72 Hz 已经成为事实上的标准。光栅扫描采用的行数也是影响视频质量的关键因素。在模拟电视中，采用的行数大约为 500 ~ 600，而计算机显示器则采用更高的行数（例

① 较传统的定义是把包含所有偶数行的场称为偶场，而把包含所有奇数行的场称为奇场。这种定义取决于第一行的标号是 0 还是 1，因此是模棱两可的。

② 帧率和行率也可分别称为垂直扫描频率和水平扫描频率。

③ 为减少闪烁的可视性，用一个旋转叶片产生 72 fps 的幻觉。

如 SVGA 显示器有 1 024 行)。帧率和行数的确定基于不同观测环境下视觉的时间和空间阈值,如 2.4 节所述。在计算机应用中需要较高的帧率和行数,以适应较短的视距和显示资料中较高频率的内容(线图形和文本)。

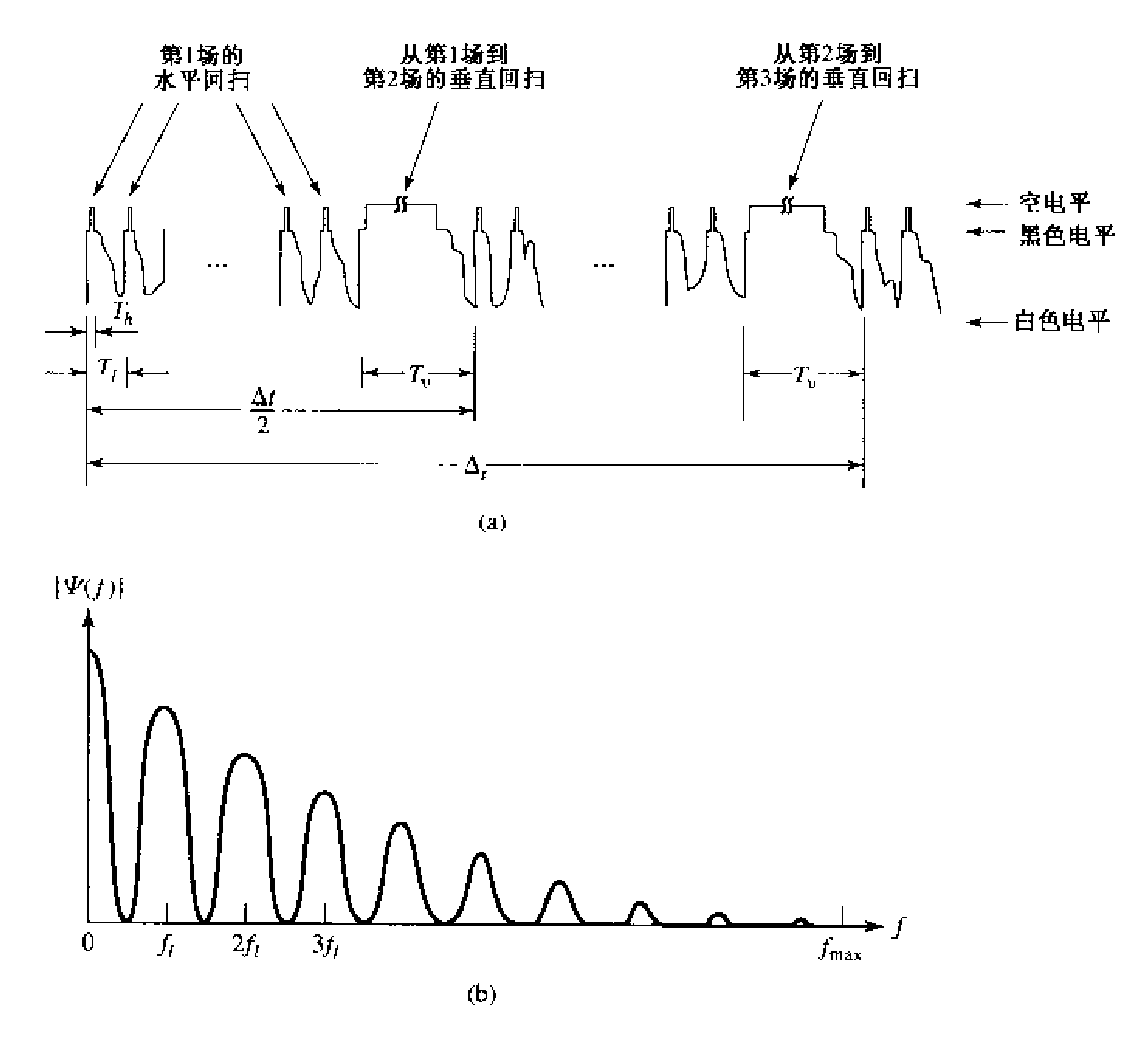


图 1.4　典型的隔行光栅扫描:(a)波形,(b)频谱

视频帧的宽对高度的比率称为图像幅型比(IAR)。例如,4:3 的 IAR 用于标准清晰度 TV(SDTV)和计算机显示,而更高的 IAR 用于宽银幕电影(高达 2.2)和 HDTV(IAR = 16:9),以产生更具戏剧性的视觉感受。

1.4　模拟彩色电视系统

在这一节,我们简要地介绍模拟电视系统,它为我们已经讨论过的许多概念提供了很好的例子。设计彩色电视系统的一个主要的限制是必须与先前的单色电视系统兼容。首先,彩色电视信号的全带宽必须符合分配给单色电视信号的带宽(在美国每个频道的带宽是 6 MHz)。其次,所有的彩色信号必须复用成一个单一的复合信号,以便单色电视接收机能从中提取亮度信号。满足以上限制的彩色电视系统的成功设计是 20 世纪一项伟大的技术发明。图 1.5 显示出包括彩色电视信号产生、传输和接收在内的主要处理步骤。下面我们简要地考察一下这些步骤。

世界上有三个不同的电视系统:NTSC 系统用于北美和包括日本在内的部分亚洲国家和地

区;PAL 系统用于大多数西欧国家和包括中国以及中东的亚洲国家;SECAM 系统用于前苏联、东欧、法国以及一些中东国家。我们将在时间和空间分辨率、彩色坐标和复用机制方面对这些系统进行比较。这里给出的资料主要来自参考文献[8,9]。关于彩色电视系统更全面的论述参见参考文献[4,15]。

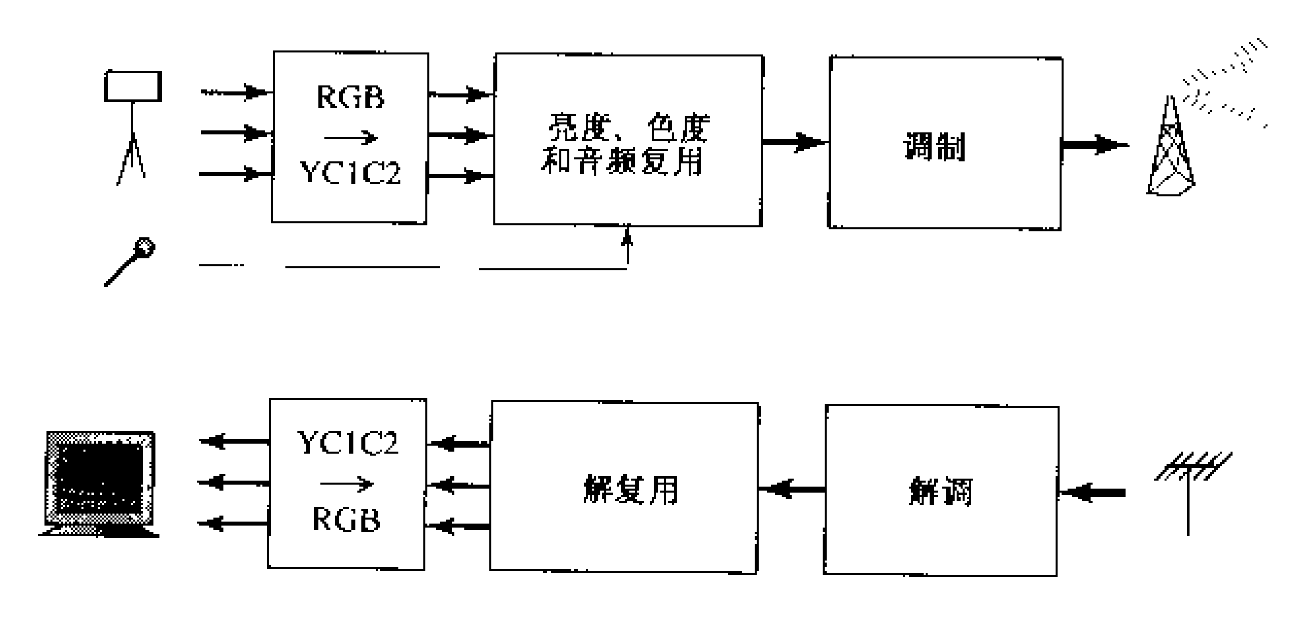


图 1.5 模拟彩色电视系统:视频产生、传输和接收

1.4.1 空间和时间分辨率

三种彩色电视系统都采用 1.3 节描述的 2:1 隔行扫描机制来摄取和显示视频图像。NTSC 系统的场率是 59.94 Hz,行数是 525 行/帧。PAL 和 SECAM 系统都采用 50 Hz 的场率,行数是 625 行/帧。选用这些场率并不干扰这些国家的标准电力系统。事实证明它们是很好的选择,因为它们与人类视觉系统的临界闪烁频率相匹配(如 2.4 节所述)。所有系统的 IAR 都是 4:3。表 1.1 概括了 NTCS,PAL 和 SECAM 视频信号的参数。对于 NTSC 系统,行间隔是 $T_l = 1/(30 \times 525) = 63.5\mu s$。但是水平回扫时间用了 $T_h = 10\ \mu s$,因此实际扫描一行的时间是 $T'_l = 53.5\ \mu s$。在相邻场间垂直回扫所用的时间是 $T_v = 1\ 333\ \mu s$,等于每场 21 个扫描行的时间。因此,有效行数是 525 - 42 = 483/帧。实际的垂直回扫只用了 9 个水平扫描行的时间。剩下的时间(12 个扫描行)是用于广播者在电视信号中传输附加数据(例如字幕、电视图文等)[①]。

表 1.1 模拟彩色电视系统的参数

参数	NTSC	PAL	SECAM
场率	59.94	50	50
行数/帧	525	625	625
行率(行/秒)	15 750	15 625	15 625
图像幅型比	4:3	4:3	4:3
彩色坐标	YIQ	YUV	YDbDr
亮度带宽(MHz)	4.2	5.0,5.5	6.0
色度带宽(MHz)	1.5(I),0.5(Q)	1.3(U,V)	1.0(U,V)
彩色副载波(MHz)	3.58	4.43	4.25(Db),4.41(Dr)
彩色调制	QAM	QAM	FM
音频副载波(MHz)	4.5	5.5,6.0	6.5
复合信号带宽(MHz)	6.0	8.0,8.5	8.0

① 摘自不同参考资料的有效行数在 480 到 495 之间变化。这里的数字是由摘自文献[4]的垂直消隐间隔计算得到的。

1.4.2 彩色坐标

用于三个电视系统的彩色坐标是不同的。为了视频摄取和显示，所有三个系统都采用RGB基色，但是对各基色光谱的定义稍微有所不同。对于视频信号的传输，为了减少所需的带宽并与单色电视系统兼容，采用了亮度/色度坐标系。下面，我们将描述用于这些系统的彩色坐标系。

在NTSC，PAL和SECAM系统中所用的彩色坐标都是从用于PAL的YUV坐标导出的，而YUV来源于XYZ坐标。根据RGB基色与YUV基色之间的关系，我们能由RGB的值确定亮度分量Y的值。两个色度值U和V分别正比于色差 $B-Y$ 和 $R-Y$，被调整到所希望的范围。特别地，YUV坐标与PAL RGB基色值的关系是：

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} \tilde{R} \\ \tilde{G} \\ \tilde{B} \end{bmatrix} \tag{1.4.1}$$

和

$$\begin{bmatrix} \tilde{R} \\ \tilde{G} \\ \tilde{B} \end{bmatrix} = \begin{bmatrix} 1.000 & 0.000 & 1.140 \\ 1.000 & -0.395 & -0.581 \\ 1.000 & 2.032 & 0.001 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix} \tag{1.4.2}$$

这里 $\tilde{R},\tilde{G},\tilde{B}$ 是归一化的伽马校正后的值，因此 $(\tilde{R},\tilde{G},\tilde{B})=(1,1,1)$，对应于在PAL/SECAM系统中定义的基准白色。

NTSC系统采用YIQ坐标，这里I和Q分量是U和V分量旋转33°后的结果。这样的旋转使I对应橙色到青色范围的彩色，Q对应绿色到紫色的范围。因为人眼对绿色到紫色范围内的变化与橙色到青色范围内的变化相比不敏感，因此Q分量可以比I分量采用更小的带宽传输[9]（这一点将在1.4.3节中详细阐述）。YIQ值与NTSC RGB系统的关系是：

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} \tilde{R} \\ \tilde{G} \\ \tilde{B} \end{bmatrix} \tag{1.4.3}$$

和

$$\begin{bmatrix} \tilde{R} \\ \tilde{G} \\ \tilde{B} \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.620 \\ 1.000 & -0.272 & -0.647 \\ 1.000 & -1.108 & 1.700 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} \tag{1.4.4}$$

对于YIQ坐标，$\tan^{-1}(Q/I)$ 近似于色调，而 $\sqrt{I^2+Q^2}/Y$ 反映饱和度。在NTSC复合视频中，I和Q分量被复用成一个信号，使得被调制信号的相位是 $\tan^{-1}(Q/I)$，而它的幅度等于 $\sqrt{I^2+Q^2}/Y$。由于传输误差对幅度的影响比对相位的影响大，因此在广播电视信号中色度信息比饱和度信息能更好地保持。这正是所希望的，因为人眼对彩色的色调更敏感。I信号与

彩色调制频率同步(在相位上),而Q信号与调制频率正交,即圆周的四分之一或90度的相位偏移,"I"和"Q"的名称由此而来(彩色复用方案将在1.4.4节中说明)。

值得注意的是,因为用于NTSC系统的RGB基色系和基准白色与用于PAL/SECAM系统的不同,在这两个系统中同样的RGB值对应的彩色是有些不同的。

SECAM系统采用YDbDr坐标,这里Db和Dr的值与U和V的值之间的关系是:

$$D_b = 3.059U, D_r = -2.169V \tag{1.4.5}$$

1.4.3 信号带宽

视频光栅的带宽可以由它的行率估计。首先,当黑色和白色行在一帧光栅中交替出现时,产生最大的垂直频率,它等于$f_{s,y}/2$(周期/像高),这里$f_{s,y}$表示有效行数。一个系统能够适当表现出来的最大频率通常低于理论极限。这个衰减因子称为凯尔(Kell)因子,记做K,它是由摄像机和显示设备的光圈函数决定的。典型电视摄像机的凯尔因子是$K = 0.7$。能够容纳的最大垂直频率与Kell因子的关系是:

$$f_{v,\max} = Kf'_{s,y}/2(\text{周期}/\text{像高}) \tag{1.4.6}$$

如果我们假设对于同样的空间距离,水平最大频率与垂直最大频率相等,则$f_{h,\max} = f_{v,\max} \cdot \mathrm{IAR}$(周期/像宽)。因为每一行都是以$T'_l$秒扫描的,所以一维光栅信号的最大频率为:

$$f_{\max} = f_{h,\max}/T'_l = \mathrm{IAR} \cdot Kf'_{s,y}/2T'_l \ \mathrm{Hz} \tag{1.4.7}$$

对于NTSC视频格式,我们有$f'_{s,y} = 483$,$T'_l = 53.5\ \mu s$。因此亮度分量的最大频率是4.2兆周期/秒或4.2 MHz。尽管色度信号的潜在带宽也许是高的,通常它也比亮度信号低得多。而且,已经发现HVS观察色度的变化有很低的阈值。一般两个色度信号被带限到很窄的带宽。正像前面所提到的,人眼对由I分量所表示的橙色到青色彩色范围内的空间变化,比由Q分量所表示的绿色到紫色范围内的空间变化更敏感。因此,I分量被带限到大约1.5 MHz,Q分量被带限到大约0.5 MHz①。表1.1列出了不同电视系统的信号带宽。

1.4.4 亮度、色度和音频的复用

为了使彩色电视信号与单色电视系统兼容,所有三个模拟电视系统都采用复合视频格式,其中三个彩色分量以及音频分量复用为一个信号。这里,我们简要地介绍NTSC所采用的机制。首先,两个色度分量$I(t)$和$Q(t)$用正交幅度调制(QAM)组合成一个信号$C(t)$。彩色的副载频选择为半行率的奇数倍,$f_c = 455\ f_l/2 = 3.58$ MHz。这样选择是为了满足如下准则:(1)在亮度分量具有很低能量的地方它应该足够高;(2)在亮度分量很强的地方它应该位于两个行率谐波的中点;(3)它应该离音频副载波足够远,音频副载波位于4.5 MHz($286f_l$),与单色电视信号相同。图1.6(a)示出了亮度和色度信号的谐波峰值是如何彼此交错的。最后,音频信号用$f_a = 4.5$ MHz的音频副载频进行频率调制(FM),并把它加到复合信号中,形成最后的复合信号。因为I分量有1.5 MHz的带宽,所以调制后色度信号的最大频率达到5.08 MHz。为了防止干扰音频信号,I信号上边带被带限到0.5 MHz。值得注意的是,I信号的下边带将进入Y信号的高端。由于这个原因,有时I信号的两个边带都被带限到0.5 MHz。如前所述,Q分量的

---

① 实际数值取决于带宽的定义,I分量从1.3到1.6 MHz变化,Q分量从0.5到0.6 MHz变化。这里的数值引自[15]。

两个边带都被带限到 0.5 MHz。最后，具有大约 4.75 MHz 带宽的整个复合信号用残留边带调制(VSB)方式调制到图像载频 f_p 上，使下边带只扩展到低于 f_p 1.25 MHz，整个信号占用6 MHz带宽。这个过程与单色电视系统是一样的。图像载波 f_p 是由广播频道决定的。图 1.6(b)示出了 NTSC 复合信号的频谱组成。表 1.1 概括了三种彩色电视系统的信号带宽和调制方式。

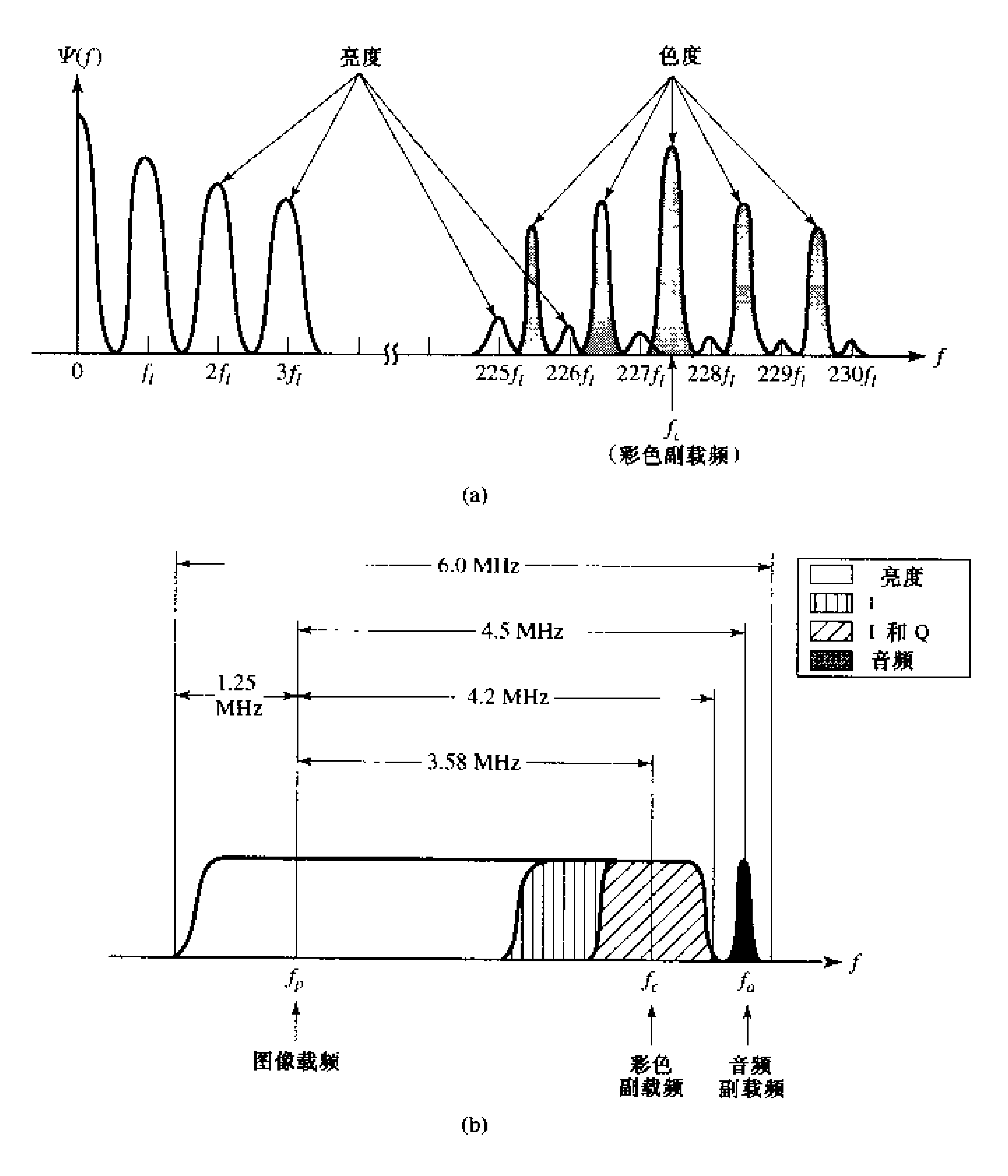


图 1.6　在 NTSC 系统中亮度、色度和音频信号的复用：(a)亮度与色度谐波之间的交错；(b)NTSC复合信号的总频谱组成

在电视接收机中，复合信号首先要解调到基带，然后音频和三个视频信号分量要解复用。为了分离视频和音频信号，可以使用低通滤波器。这个过程在单色和彩色电视中是一样的。为了进一步从亮度信号中分离出色度信号，理想情况下，应该使用梳状滤波器，以便利用这两个信号中谐波频率交错的优点。大多数高级电视机采用在色度分量相应的谐波上为零频的数字梳状滤波器来实现这种滤波。然而，低级电视机采用简单的 RC 电路实现截止频率为 3 MHz 的低通滤波，这会使在提取出的亮度信号中残留不希望的 I 分量，反过来也一样。这将导致彩

色串扰和亮度串扰的人工痕迹。彩色串扰是指由于高频亮度信号接近彩色副载频而产生的假彩色。亮度串扰是指由已调色度信息导致的错误的高频边缘图案(对于各种滤波器效应的一个很好的图示参见文献[1])。提取出色度信号后,用相应的彩色解调方式分离出两个色度分量。最后,将三个彩色分量转换到 RGB 坐标用于显示。

1.4.5 模拟视频录像

随着模拟电视系统的发展,已经开发了各种视频磁带录像(VTR)技术,以便于专业视频生产(录像和编辑),以及消费者的录像(家用视频)和放像(VCR)。表 1.2 概括了通用视频磁带格式。

表 1.2 模拟视频磁带格式

视频格式	磁带格式	水平行	亮度带宽	应用
复合	VHS,8 mm	240	3.0 MHz	消费
	Umatic SP	330	4.0 MHz	专业
S 视频	S-VHS,Hi8	400	5.0 MHz	高质量消费
分量	Betacam SP	480	4.5 MHz	专业

1.5 数字视频

1.5.1 符号

数字视频可以通过采样光栅扫描或直接用数字视频摄像机获得。目前,所有的数字摄像机都使用 CCD 传感器。与模拟摄像机一样,数字摄像机把成像景物采样成离散的帧。每一帧由水平和垂直都离散化的 CCD 阵列的输出值组成。数字视频是由帧率 $f_{s,t}$,行数 $f_{s,y}$ 和每行的样点数 $f_{s,x}$ 定义的。根据这些参数,可以求出时间采样间隔或帧间隔 $\Delta_t = 1/f_{s,t}$,垂直采样间隔 Δ_y = 像高$/f_{s,y}$,水平采样间隔 Δ_x = 像宽$/f_{s,x}$。在本书中,我们用$\psi(m,n,k)$表示数字视频,这里整数标号 m 和 n 是列标号和行标号,k 是帧数。相应于这些整数标号的实际的空间和时间位置是:$x = m\Delta_x, y = n\Delta_y, t = k\Delta_t$。为了方便,我们用$\psi(x,y,t)$表示通常意义的视频信号,它可以是模拟的或数字的。只有当特别指出是数字视频时才用$\psi(m,n,k)$。

除了前面的参数,数字视频另一个重要的参数是用于表示一个像素值(只是亮度,或是三个彩色值)的比特数,并把它记为 N_b。传统上,用 8 比特或 256 个级指定亮度和/或三个彩色值的每一个彩色。因此,对于单色电视 $N_b = 8$,而对于彩色电视 $N_b = 24$。数字视频的码率由 $R = f_{s,t}f_{s,x}f_{s,y}N_b$确定,单位是比特/秒(或 bps,通常的单位是千比特/秒(kbps)或兆比特/秒(Mbps))。一般地说,对于视频信号的亮度和色度分量,时间和空间采样率可以不同。在这种情况下,N_b 应该反映在亮度采样分辨率下用于每个像素的等效比特数。例如,如果每个色度分量的水平和垂直采样率都是亮度采样率的一半,那么每 4 个 Y 采样有两个色度采样。如果每个采样用 8 比特表示,那么在 Y 分辨率下每个采样的等效比特数为:$(4\times8+2\times8)/4 = 12$ 比特。

当在监视器上显示数字视频时,每个像素被表示为具有指定给该像素的一种恒定彩色的一个矩形区域。这个矩形区域的宽度对高度的比率称为像素幅型比(PAR)。它与显示区域的

IAR 以及图像尺寸的关系为：

$$PAR = IAR \cdot f_{s,y}/f_{s,x} \tag{1.5.1}$$

为了正确显示数字化视频信号，必须在指定 $f_{s,x}$ 和 $f_{s,y}$ 的同时指定 PAR 或 IAR。显示设备应该与为该信号指定的（或由指定的 IAR 导出的）PAR 一致。否则，物体的外形将会失真。例如，如果显示 PAR 大于指定的 PAR，那么一个人的图像会变得较胖和较矮。在计算机行业中，通常采用的 PAR 为 1.0。另一方面，在电视行业中，由于历史的原因使用非方形像素，其原因将在 1.5.2 节进行解释。

1.5.2 ITU-R BT.601 数字视频

BT.601 信号的空间分辨率　致力于将广播质量的不同的模拟电视视频信号所使用的数字格式标准化，国际电信联盟-无线电部门（ITU-R）提出了 BT.601 建议[35]。这个标准指定了幅型比为 4:3 和 16:9 的数字视频格式，我们仅讨论幅型比为 4:3 的格式①。为了把光栅扫描信号转换成数字视频信号，只需采样一维波形。若每行取 $f_{s,x}$ 采样的样点总数，则等效采样率为 $f_s = f_{s,x}f_{s,y}f_{s,t} = f_{s,x}f_l$ 样点/秒。在 BT.601 标准中，选择的采样频率要满足两个约束条件：(1) 水平采样分辨率要尽可能与垂直采样分辨率匹配，即令 $\Delta_x \approx \Delta_y$；(2) 同样的采样率应该用于 NTSC 和 PAL/SECAM 系统，而且应该是这些系统中各个行率的倍数。第一个准则要求 $f_{s,x} \approx IAR \cdot f_{s,y}$ 或者是 $f_s \approx IAR \cdot f_{s,y}^2 f_{s,t}$，这导致对于 NTSC 和 PAL/SECAM 系统 f_s 分别为 11 MHz 和 13 MHz。从而一个既接近这两个数又满足第二个准则的数便可以选出，这个数是

$$f_s = 858f_l(\text{NTSC}) = 864f_l(\text{PAL/SECAM}) = 13.5\ \text{MHz} \tag{1.5.2}$$

对于 NTSC，每行的像素数为 $f_{s,x} = 858$，对于 PAL/SECAM 为 $f_{s,x} = 864$。这两种格式分别称为 525/60 和 625/50 信号，如图 1.7 所示。在 525 行和 625 行系统中有效行数分别是 $f'_{s,y} = 480$ 和 $f'_{s,y} = 576$，但是每行的有效像素数是相同的，均为 $f'_{s,x} = 720$ 像素。剩下是在水平和垂直回扫中获得的样点，它们落入无效区。

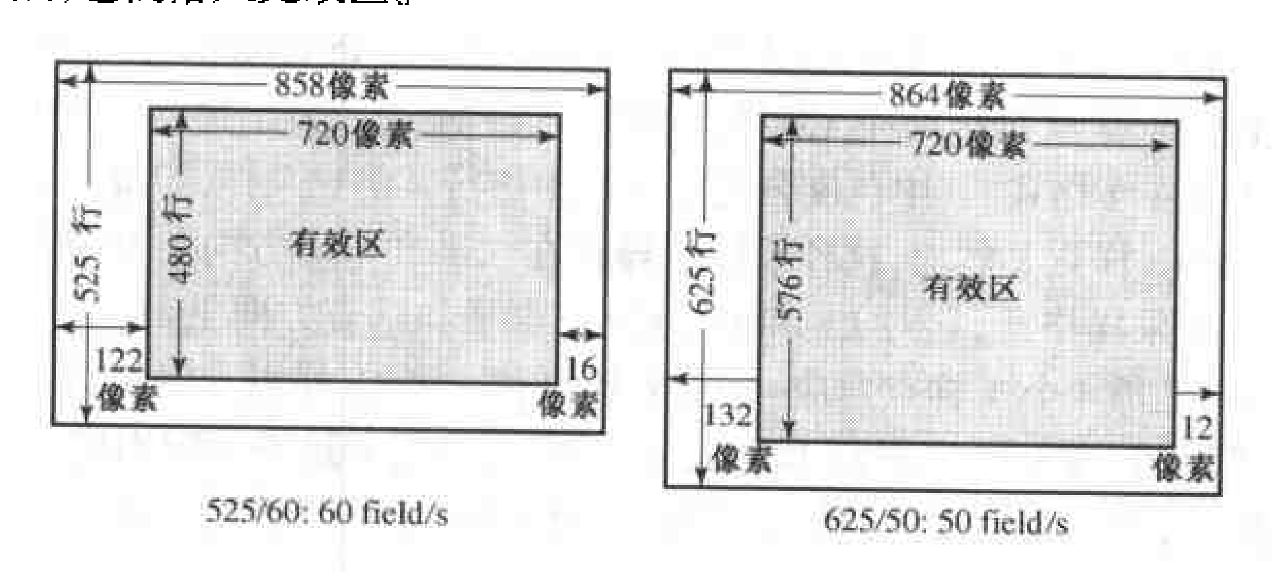


图 1.7　BT.601 视频格式

对于 BT.601 信号，像素宽对高的比率并不是 1，即像素的物理区域并不是一个方块。特别地，对于 525/60 信号，$PAR = \Delta_x/\Delta_y = IAR \cdot f'_{s,y}/f'_{s,x} = 8/9$，对于 625/50 信号则为 16/15。为了显示 BT.601 信号，显示设备应该有适当的 PAR，否则图像将失真。例如，当在 PAR 为 1 的计算

① ITU-R 以前称为国际无线电咨询委员会（CCIR），并且 BT.601 格式的 4:3 幅型比的版本被称为 CCIR601 格式。

机屏幕上显示时,525/60 信号将在水平方向被拉长,而 625/50 信号将在垂直方向被拉长。理想情况下,应该对原始信号重采样,以使 $f'_{s,x} = \text{IAR} \cdot f'_{s,y}$。例如,525/60 和 625/50 信号应该分别被重采样为每行 640 和 768 个有效像素。

彩色坐标和色度的亚采样 除了图像分辨率外,BT.601 建议还定义了一个数字彩色坐标,称为 YCbCr。Y,Cb 和 Cr 分量是模拟 Y,U 和 V 分量的伸缩和移位形式,这里引入伸缩和移位运算以便使产生的分量在(0~255)范围内取值。关于这个彩色坐标设计的更详细的解释,读者可参阅参考文献[8];这里我们只给出了由数字 RGB 坐标导出这个坐标的变换矩阵。假设 RGB 值的范围是(0~255),YCbCr 值与 RGB 值的关系为:

$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} \tag{1.5.3}$$

相反的关系为:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.164 & 0.000 & 1.596 \\ 1.164 & -0.392 & -0.813 \\ 1.164 & 2.017 & 0.000 \end{bmatrix} \begin{bmatrix} Y - 16 \\ C_b - 128 \\ C_r - 128 \end{bmatrix} \tag{1.5.4}$$

在上述关系式中,$R = 255\tilde{R}$, $G = 255\tilde{G}$, $B = 255\tilde{B}$ 是在 NTSC 和 PAL/SECAM 系统中定义的归一化 RGB 基色 $\tilde{R}, \tilde{G}, \tilde{B}$ 的数字等效值。在 YCbCr 坐标中,Y 反映亮度,伸缩后的范围是(16~235);C_b 和 C_r 分别是色差 $B - Y$ 和 $R - Y$ 伸缩后的形式。通过伸缩和移位使它们的取值范围为(16~240)。C_r 的最大值对应红色($C_r = 240$ 或者 $R = 255, G = B = 0$),而最小值对应青色($C_r = 16$ 或者 $R = 0, G = B = 255$)。类似地,C_b 的最大和最小值分别对应蓝色($C_b = 240$ 或者 $R = G = 0, B = 255$)和黄色($C_b = 16$ 或者 $R = G = 255, B = 0$)。

前面介绍的空间采样率指的是亮度分量 Y。对于色度分量 Cb 和 Cr,通常只用这个采样率的一半,即 $f_{s,c} = f_s/2$。这使得每行的像素数减少一半,但每帧的行数相同。这被称为 4:2:2 格式,意味着每 4 个 Y 样点对应两个 Cb 样点和两个 Cr 样点。为了进一步降低所需的码率,BT.601还定义了 4:1:1 格式,其中色度分量在每行中进行 4 倍亚采样,即每 4 个 Y 样点对应一个 Cb 样点和一个 Cr 样点。然而,这种采样法导致在水平和垂直方向的分辨率很不对称。因此又提出了另一种采样格式,它对 Cb 和 Cr 分量在水平和垂直方向上都进行一半的亚采样。在这种格式中,还是每 4 个 Y 样点对应一个 Cb 样点和一个 Cr 样点。但是为了避免与前面定义的 4:1:1 格式混淆,这种格式称为 4:2:0。对于需要很高分辨率的应用场合,定义了 4:4:4 格式,它以与亮度分量完全一样的分辨率采样色度分量。不同格式的亮度和色度样点的相对位置如图 1.8 所示[①]。

在第 4 章中,我们将讨论以不同的时间/空间分辨率转换视频信号的解决办法。不同的彩色亚采样格式间的转换是该章的一个练习题。

BT.601 信号的原码率是由彩色的亚采样因子决定的。对于最通用的 4:2:2 格式,每两个 Y 采样对应两个色度采样,每个采样用 8 比特表示。因此,对于每个 Y 采样,等效的比特率为

---

① 对于 4:2:0 格式,Cr 和 Cb 样点也可以位于 4 个对应的 Y 样点的中心,如图 13.14 所示。

$N_b = 16$ 比特，原码率是 $f_s N_b = 216$ Mbps。原码率对应的有效区是 $f_{s,t} f'_{s,y} f'_{s,x} N_b = 166$ Mbps。对于 4:2:0 格式，每 4 个 Y 采样对应两个色度采样，对于每个 Y 采样等效的比特率是 $N_b = 12$ 比特。因此原码率为 162 Mbps，在有效区为 124 Mbps。对于 4:4:4 格式，对于每个 Y 采样等效的比特率为 $N_b = 24$ 比特，原码率为 324 Mbps，而有效区为 249 Mbps。表 1.3 概括了不同的 BT.601信号的分辨率和码率。

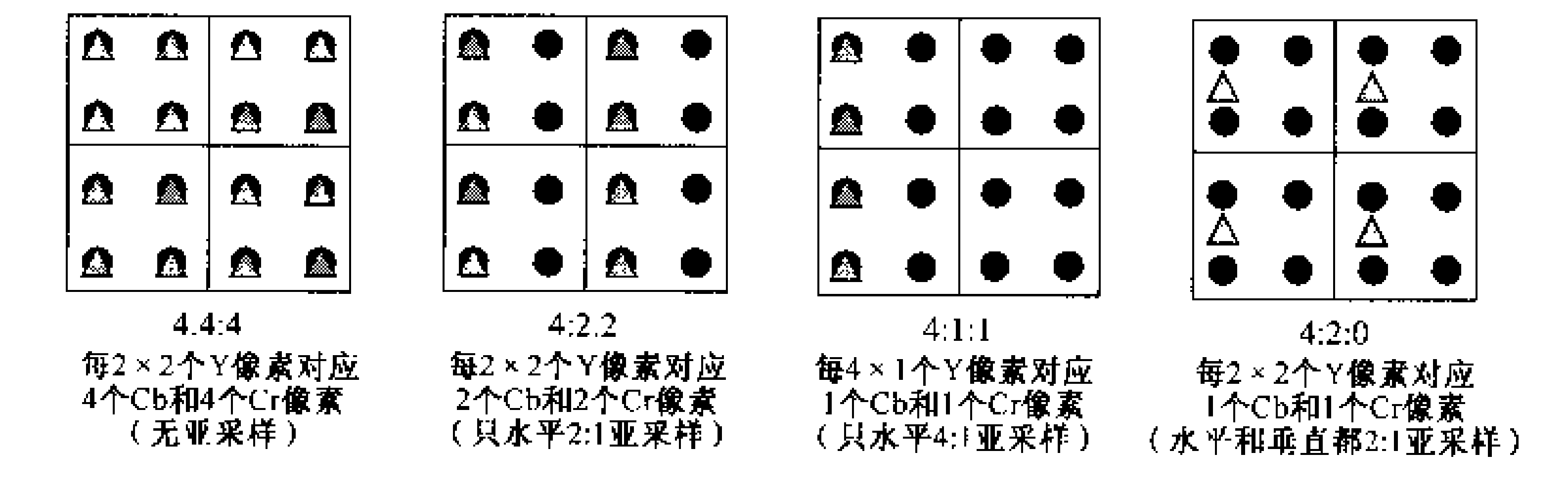


图 1.8　BT.601 色度亚采样格式。注意在任何一个格式中两个相邻的行属于两个不同的场

表 1.3　不同应用的数字视频格式

视频格式	Y 尺寸	彩色采样	帧率	原码率(Mbps)
地面、有线及卫星 HDTV，MPEG-2 视频，20 ~ 45 Mbps				
SMPTE 296M	1280 × 720	4:2:0	24P/30P/60P	265/332/664
SMPTE 295M	1920 × 1080	4:2:0	24P/30P/60I	597/746/746
视频制作，MPEG-2，15 ~ 50 Mbps				
BT.601	720 × 480/576	4:4:4	60I/50I	249
BT.601	720 × 480/576	4:2:2	60I/50I	166
高质量视频发布(DVD，SDTV)，MPEG-2，4 ~ 8 Mbps				
BT.601	720 × 480/576	4:2:0	60I/50I	124
中质量视频发布(VCD，WWW)，MPEG-1，1.5 Mbps				
SIF	352 × 240/288	4:2:0	30P/25P	30
ISDN/因特网视频会议，H.261/H.263，128 ~ 384 kbps				
CIF	352 × 288	4:2:0	30P	37
有线/无线调制解调可视电话，H.263，20 ~ 64 kbps				
QCIF	176 × 144	4:2:0	30P	9.1

BT.601 格式用于高质量的数字视频场合，一般 4:4:4 和 4:2:2 格式用于视频制作和编辑，4:2:0 格式用于视频发行(如数字视盘(DVD)上的电影，视频点播(VOD)等)。MPEG-2[①] 视频压缩标准主要是为压缩 BT.601 4:2:0 信号开发的，尽管它也能处理较高或较低分辨率的视频信号。原有效码率为 124 Mbps 的典型 4:2:0 信号可以被压缩到大约 4 ~ 8 Mbps。我们将在

① “MPEG”代表国际标准化组织或 ISO 的运动图像专家组。

13.5 节中介绍 MPEG-2 视频编码算法。

1.5.3 其他数字视频格式和应用

除了 BT.601 格式,还定义了几个其他的标准数字视频格式。表 1.3 概括了这些视频格式以及它们的主要应用、压缩方法和压缩后的比特率。国际电信联盟-电信部门(ITU-T)规定的 CIF(通用媒体格式)在水平和垂直维上的分辨率都大约是 BT.601 4:2:0 信号的一半,它是为视频会议应用而开发的;在水平和垂直维上具有 CIF 一半分辨率的 QCIF 用于可视电话及类似的应用场合。它们都是非隔行的。ITU-T H.261 编码标准的推出是为了以这些格式将视频信号压缩到 $p \times 64$ kbps,其中 $p = 1,2,\cdots,30$,以便在只允许传输速率是 64 kbps 整数倍的 ISDN(综合业务数字网)线路上传输。一般原码率为 37.3 Mbps 的 CIF 信号能被压缩到大约 128 ~ 384 kbps,具有较好的质量,而原码率为 9.3 Mbps 的 QCIF 信号能被压缩到 64 ~ 128 kbps。其后的 H.263 标准可以在同样的比特率下达到比 H.261 更好的质量。例如,H.263 能把一幅 QCIF 图像压缩到大约 20 kbps,而与 64 kbps 码率下的 H.261 相比质量相当或更好。这使可视电话通过 28.8 kbps 的调制解调器线路通信成为可能。

与 ITU-T 的努力并行,ISO-MPEG 也规定了一系列数字视频标准。SIF(源媒体格式)有效区尺寸实际上是 BT.601 信号的四分之一,大致与 CIF 相当。这种格式的目标定位于中等质量的视频应用,如视频游戏和 CD 电影之类。类似 BT.601,有两种 SIF 格式:一种的帧率是 30 Hz,行数是 240,另一种的帧率是 25 Hz,行数是 288,它们都是 352 像素/行。还有一个相应的 SIF-I 格式系列,它们是 2:1 隔行的。MPEG-1 算法能够把一个原码率为 30 Mbps 的典型的 SIF 视频信号压缩到大约 1.1 Mbps,质量类似于在 VHS VCR 上看到的分辨率,低于广播电视的质量。1.1 Mbps的码率能回放存取速率为 1.5 Mbps 的 CD-ROM 上的数字电影。在视频 CD(VCD)上发布 MPEG-1 电影是 20 世纪 90 年代早期数字视频强化消费者市场的标志。在 20 世纪 90 年代中期开始的基于 MPEG-2 的 DVD,开辟了高质量数字视频娱乐的新时代。MPEG-2 技术也是下一代采用数字压缩和传输技术的全数字电视系统的奠基石。表 1.3 详细列出了所讨论的视频格式,以及它们的主要应用、压缩方法和压缩后的比特率(第 13 章将给出压缩标准的更多内容)。

BT.601 格式是用于数字电视(DTV)的标准图像格式。为了进一步增强视频质量,几个 HDTV 格式也已经被运动图像和电视工程师协会(SMPTE)进行了标准化,它们也列入表 1.3 中。HDTV 的一个显著特征是它的幅型比较宽,为 16:9,而 SDTV 是 4:3。在水平和垂直维上都具有两倍到三倍的图像分辨率,而且为降低隔行效应采用了逐行扫描。MPEG-2 视频压缩标准已经推出一个高类(profile)用于压缩 HDTV 视频。典型情况下它能把码率压缩到大约 20 Mbps,同时保持很高的所需要的质量。选择这样的视频比特率,是为了采用数字调制技术传输时其与音频结合的比特流仍然能够载入到 6 MHz 的地面频道,这是美国为 HDTV 分配的频道带宽。有关 DTV 的更多信息,读者可参考 Whitaker [14]。

1.5.4 数字视频录像

为了以数字格式存储视频,已推出各种数字视频磁带录像机(DVTR)格式,它们在视频格式运用、纠错编码技术以及存储密度等方面有所不同。表 1.4 列出了一些标准和专有的磁带格式。D1 ~ D5 是以原始的非压缩的格式存储视频,而其他格式要对视频进行预压缩。只使用

了保守的压缩量，因此图像质量的下降不会低于预期应用可接受的程度。参考文献[12]中对 DVTR 做了一个很好的综述。关于 DVTR 磁记录和操作基础物理学的广泛的内容可参阅参考文献[13]。

除了磁带录像机，VCD 和 DVD 是使用光盘的两种视频存储设备。利用 MPEG-1 和 MPEG-2 技术能以较高质量分别存储 SIF 和 BT.601 视频，目前，VCD 和 DVD 是只读的，因此它们主要用于预先记录的视频的发布，而不是消费者记录视频的工具。

如参考文献[11]和[10]所介绍的基于硬盘的视频录像系统也已经出现。这些系统使消费者能把高达 30 小时的广播电视节目以 MPEG-2 压缩格式记录到硬盘上，以便以后能像普通 VCR 录像机一样用快进、慢动作等功能观看。它们还允许即时暂停正在观看的活动节目，把活动的视频在暂停期间存储到磁盘上。随着硬盘价格的持续下降，硬盘 DVTR 可能最终会取代速度慢而且存储容量小的磁带录像系统。

表 1.4　数字视频磁带格式

磁带格式	视频格式	源码率	压缩率	压缩方法	应用
非压缩格式					
SMPTE D1	BT.601 4:2:2	216 Mbps	N/A	N/A	专业人员
SMPTE D2	BT.601	114 Mbps	N/A	N/A	专业人员
SMPTE D3	BT.601	114 Mbps	N/A	N/A	专业人员/消费者
SMPTE D5	BT.601 4:2:2(10 bit)	270 Mbps	N/A	N/A	专业人员
压缩格式					
Digital Betacam	BT.601 4:2:2	166 Mbps	80 Mbps	Frame DCT	专业人员
Betacam SX	BT.601 4:2:2	166 Mbps	18 Mbps	MPEG-2 (仅 I 和 B 模式)	消费者
DVCPRO50	BT.601 4:2:2	166 Mbps	50 Mbps	帧/场 DCT	专业人员
DVCPRO25(DV)	BT.601 4:1:1	124 Mbps	25 Mbps	帧/场 DCT	消费者

1.5.5　视频质量的测量

为了指导视频处理，有必要定义一个能测量原始信号与处理后信号之间的差别的客观准则。这在视频编码应用中是尤其重要的，这时我们必须测量由压缩引起的失真。理想情况下，这种测量应该与两个视频序列之间感觉上的差异相关联。然而，找到这样一种方法是一件非常困难的工作。尽管已经提出了各种质量测量，然而与人的视觉感觉有好的相关性的方法在计算上是相当复杂的。目前大多数视频处理系统的设计是使两个视频序列 ψ_1 和 ψ_2 之间的均方误差（MSE）最小化，它定义为：

$$\mathrm{MSE} = \sigma_e^2 = \frac{1}{N}\sum_{k}\sum_{m,n}(\psi_1(m,n,k) - \psi_2(m,n,k))^2 \tag{1.5.5}$$

这里 N 是每个序列中的总像素数。对于彩色视频，每个彩色分量的 MSE 是分别计算的。

代替 MSE，以分贝（dB）为单位的峰值信噪比（PSNR）更经常用于视频编码中的质量测量。PSNR 定义为：

$$\mathrm{PSNR} = 10\log_{10}\frac{\psi_{\max}^2}{\sigma_e^2} \tag{1.5.6}$$

这里 $\psi_{\max}$ 是视频信号的峰（最大）强度值。对于最通常的每彩色 8 比特视频，$\psi_{\max} = 255$。注意，对于固定的峰值，PSNR 完全由 MSE 决定。PSNR 比 MSE 更经常使用，是因为人们总是倾向于

将图像质量与某种 PSNR 范围相联系。作为一个主要的准则，对于亮度分量，一个高于 40 dB 的 PSNR 一般意味着一个极好的图像(即与原始图像很接近)，30 ~ 40 dB 之间通常意味着一个好的图像(即失真可觉察，但可以接受)，20 ~ 30 dB 之间是相当差的，最后，PSNR 低于 20 dB 是不可接受的。

计算两个序列间的 PSNR 是不值得的，在每两个相应帧间计算 PSNR，然后对各个帧所得到的值取平均是不正确的。正确的是，我们应该在相应帧间计算 MSE，对所有帧得到的 MSE 值取平均、最后把 MSE 转换到 PSNR。

主要为了减少计算量，有时用来代替 MSE 的一种测量是平均绝对误差(MAD)，定义为

$$\mathrm{MAD} = \frac{1}{N}\sum_{k}\sum_{m,n}|\psi_1(m,n,k) - \psi_2(m,n,k)| \tag{1.5.7}$$

例如，对于运动估计，MAD 经常用于为当前帧的已知块在另一帧中寻找到最佳的匹配块。

众所周知，MSE 和 PSNR 与图像间的感觉失真的相关性不是很好。但是这些测量方法几乎无例外地用于图像和视频编码、运动补偿预测以及图像恢复的客观失真测量，这一方面由于它们在数学上易处理，另一方面由于缺乏更好的代替方法。设计容易计算的并且与感觉失真相关性很好的客观失真测量方法仍是一个开放的研究课题。在本书中，我们主要用 MSE 和 PSNR 作为失真测量。

1.6 小结

彩色的产生、感觉和规定(1.1 节)

- 光的彩色决定于频谱成分。任何彩色都能通过三基色的混合而产生。最通用的基色系包括红色、绿色和蓝色。
- 人眼是通过视网膜上调谐于红色、绿色和蓝色波长的接收细胞(锥状细胞)感知彩色的。彩色可用三个属性加以描述：亮度(明亮度)、色调(彩色基调)和饱和度(彩色纯度)。人眼对亮度最敏感，然后是色调，最后是饱和度。
- 彩色可以由三个数值规定：或者是相应于三基色的贡献值(即三激励值)，或者是一个亮度值和两个色度值。

模拟视频(1.3 节)

- 用于广播电视、视频摄像机、视频显示等的模拟视频信号以光栅扫描格式存储。光栅扫描的视觉质量和带宽是由它的帧率和行数决定的。
- 隔行扫描是用牺牲垂直分辨率换取增强的时间分辨率的一种机制。然而，它也会导致隔行效应。

模拟彩色电视系统(1.4 节)

- 现在世界上采用三种模拟彩色电视系统：NTSC，PAL 和 SECAM。它们都采用 2:1 隔行扫描，但是在帧率、行数、彩色坐标以及亮度和色度的复用方面不同。
- 在彩色电视系统中，亮度和两个色度分量以及伴音信号都采用调制技术(频率偏移)复用成一个复合信号，复用方式的设计使得彩色电视系统与单色电视系统向下兼容。而

且,各个分量调制频率的选择使它们之间的干扰最小化。

数字视频(1.5 节)

- BT.601 是由采样模拟彩色电视信号得到的一种数字视频格式。采样率的选择使水平和垂直采样率近似,并且使 NTSC 和 PAL/SECAM 系统的码率相同。
- 色度分量可以用低于亮度分量的速率采样。BT.601 定义了不同的彩色亚采样格式。
- 为了降低数字视频信号的源码率,压缩是必要的,并因此会降低存储和传输的成本。已经为不同的视频应用研制出了不同的视频压缩标准。

1.7　习题

1.1　叙述人类感知彩色的机制。

1.2　如果一束光在相应于红色、绿色和蓝色的频率上有近似相同的能量,而在其他频率的能量为零,你所感受到的彩色是什么? 如果只有红色和绿色频率呢?

1.3　如果将红色、绿色和蓝色的染料以相等的比例混合,你所感受到的彩色是什么? 只混合红色和绿色的染料呢?

1.4　对于 RGB 坐标中的下列彩色,分别确定它们在 YIQ 和 YUV 坐标中的值。

(a) (1,1,1)　(b) (0,1,0)　(c) (1,1,0)　(d) (0,1,1)

1.5　对于数字 RGB 坐标中的下列彩色,确定它们在 YCbCr 坐标中的值。

(a) (255,255,255)　(b) (0,255,0)　(c) (255,255,0)　(d) (0,255,255)

1.6　在 1.5.2 节中,我们说 Cr 的最大值对应红色,而最小值对应青色,类似地,Cb 的最大和最小值分别对应蓝色和黄色。用 YCbCr 到 RGB 的坐标变换验证这些表述。

1.7　在图 1.4 中,我们给出了典型光栅信号的频谱。为什么视频信号的频谱是接近周期性的? 谐波波瓣的宽度是由什么决定的?

1.8　逐行与隔行扫描各有什么优缺点? 对于每帧同样的行数,逐行扫描光栅的最大时间频率与把每帧分成两场的隔行扫描的最大时间频率之间的关系是什么? 最大垂直频率之间的关系又怎样?

1.9　在 1.4.3 节中,我们根据 NTSC 信号的扫描参数估计了它的带宽。用同样的方法估计 PAL 和 SECAM 信号的带宽。

1.10　叙述形成一个复合彩色视频信号的过程。怎么选择彩色和音频的副载波频率?

1.11　采用分量与复合格式各有什么优缺点?

1.12　自修项目:用示波器(a)画出波形,并 (b)测量电视机或录像机输出的复合视频信号的频谱。

1.13　自修项目:用 A/D 转换器将一个复合视频信号数字化,然后用 MATLAB 确定其频谱。并且,通过滤波分离亮度、色度和音频信号。

1.8 文献目录

[1] Blinn, J. F. NTSC: nice technology, super color. *IEEE Computer Graphics and Applications Magazine* (March 1993):17–23.

[2] Boynton, R. M. *Human Color Vision*. New York: Holt, Rinhart, and Winston, 1979.

[3] Gonzalez, R. G., and R. E. Woods. *Digital Image Processing*. 2nd ed. Reading, MA: Addison–Wesley, 1992.

[4] Grob, B., and C. E. Herndon. *Basic Television and Video Systems*. 6th ed. New York: McGraw Hill, 1999.

[5] Hashimoto, Y., M. Yamamoto, and T. Asaida. Cameras and display systems. *IEEE* (July 1995), 83(7):1032–43.

[6] Haskell, B. G., A. Puri, and A. N. Netravali. *Digital Video: An Introduction to MPEG-2*. New York: Chapman & Hall. 1997.

[7] ITU-R. BT.601-5: Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios, 1998. (Formerly CCIR601).

[8] Netravali, A. N., and B. G. Haskell. *Digital Pictures—Representation, Compression and Standards*. 2nd ed. New York: Plenum Press, 1995.

[9] Pritchard, D. H. US color television fundamentals. *IEEE Trans. Consum. Electron.* (1977), CE-23(4):467–78.

[10] ReplayTV, Inc.: http://www.replaytv.com

[11] TiVo, Inc.: http://www.tivo.com

[12] Umemoto, M., Y. Eto, and T. Fukinuki. Digital video recording. *IEEE* (July 1995), 83(7):1044–54.

[13] Watkinson, J. *The Art of Digital Video*. 2nd ed. Oxford: Focal Press, 1994.

[14] Whitaker, J. C. *DTV Handbook: The Revolution in Digital Video*. 3rd ed. New York: McGraw Hill, 2001.

[15] Whitaker J. C. and K. B. Benson. ed. *Standard Handbook of Video and Television Engineering*. 3rd ed. New York: McGraw-Hill, 2000.

[16] Wyszecki, G., and W. S. Stiles. *Color Science*. New York: John Wiley, 1967.

[17] Young, T. On the theory of light and colors. *Philosophical Transactions of the Royal Society of London* (1802), 92:20–71.

第2章 视频信号的傅里叶分析和人类视觉系统的频率响应

傅里叶分析是信号分析的重要工具。我们假设读者对一维和二维空间的傅里叶变换以及用这种变换进行信号处理的工具是熟悉的。在本章中,我们首先将这些结果推广到 K 维(K-D),这里 K 可以是任何正整数,然后集中于它们在三维视频信号中的应用。我们将揭示空间和时间频率的意义,以及它们的相互关系。最后讨论对不同频率分量的视觉敏感性。

2.1 多维连续空间信号和系统

多数用于多维信号与系统的定理和技术是一维和二维信号与系统的直接推广。在本节中,我们介绍在 K 维实空间$\mathcal{R}^K = \{[x_1, x_2, \cdots, x_K]^T \mid x_k \in \mathcal{R}, k \in \mathcal{K}\}$中一些信号分析的重要概念和定理,这里$\mathcal{R}$是实数集,而$\mathcal{K} = \{1, 2, \cdots, K\}$。我们从定义 K 维信号、K 维信号间通常的运算以及特殊的 K 维信号开始,然后定义 K 维信号的傅里叶变换表示。最后,我们定义 K 维系统和线性移不变系统的特性。叙述有意识地保持简洁,我们也有意识地避免讨论各种积分公式的收敛条件。这个课题更详尽的论述可参阅参考文献[2]。

定义 2.1 设 K 维连续空间信号$\psi(\mathbf{x})$是 K 维连续变量 $\mathbf{x} = [x_1, x_2, \cdots, x_K] \in \mathcal{R}^K$ 的函数。这个函数能取任何值,实数和复数,如果取实数就称为实函数。

信号可以按照矢量空间的加法和比例运算进行组合。K 维函数之间的另一个重要运算是卷积,如下面的定义所示。

定义 2.2 两个定义在空间$\mathcal{R}^K$ 上的信号$\psi(\mathbf{x})$和 $h(\mathbf{x})$的卷积定义为:

$$\psi(\mathbf{x}) * h(\mathbf{x}) = \int_{\mathcal{R}^K} \psi(\mathbf{x} - \mathbf{y}) h(\mathbf{y}) d\mathbf{y} \tag{2.1.1}$$

在几个特殊的函数中,δ 函数(也称为冲激函数)在表征信号与线性系统中起着很重要的作用。

定义 2.3 $\mathcal{R}^K$ 中的 δ 函数满足:

$$\delta(\mathbf{x}) = \begin{cases} \infty, \mathbf{x} = \mathbf{0} \\ 0, \text{ 其他} \end{cases} \quad \text{和} \quad \int_{\mathcal{R}^K} \delta(\mathbf{x}) d\mathbf{x} = 1 \tag{2.1.2}$$

函数的性质 下面是几个与 δ 函数有关的重要恒等式,这里$\psi(\cdot)$是任意的 K 维函数:

$$\psi(\mathbf{x}) * \delta(\mathbf{x} - \mathbf{x}_0) = \psi(\mathbf{x} - \mathbf{x}_0) \tag{2.1.3}$$

$$\int_{\mathcal{R}^K} \psi(\mathbf{x}) \delta(\mathbf{x} - \mathbf{x}_0) d\mathbf{x} = \psi(\mathbf{x}_0) \tag{2.1.4}$$

$$\int_{\mathcal{R}^K} \exp(j2\pi \mathbf{f}_0^T \mathbf{x}) d\mathbf{x} = \delta(\mathbf{f}_0) \tag{2.1.5}$$

定义 2.4（连续空间傅里叶变换） 信号$\psi(\mathbf{x})$的连续空间傅里叶变换(CSFT)[①] 定义为[②]：

$$\Psi_c(\mathbf{f}) = \int_{\mathcal{R}^K} \psi(\mathbf{x})\exp(-j2\pi\mathbf{f}^T\mathbf{x})\,d\mathbf{x} \tag{2.1.6}$$

这里 $\mathbf{f}=[f_1,f_2,\cdots,f_K]^T\in\mathcal{R}^K$ 表示连续频率域变量。

定理 2.1（逆连续空间傅里叶变换） 一个函数可以由它的 CSFT 通过下式得到：

$$\psi(\mathbf{x}) = \int_{\mathcal{R}^K} \Psi_c(\mathbf{f})\exp(j2\pi\mathbf{f}^T\mathbf{x})\,d\mathbf{f} \tag{2.1.7}$$

为了证明前面的定理，把公式(2.1.6)代入公式(2.1.7)的右边，我们得到：

$$\int_{\mathcal{R}^K} \Psi_c(\mathbf{f})\exp(j2\pi\mathbf{f}^T\mathbf{x})\,d\mathbf{f} = \int_{\mathcal{R}^K} \psi(\mathbf{y})\int_{\mathcal{R}^K}\exp(j2\pi\mathbf{f}^T(\mathbf{x}-\mathbf{y}))\,d\mathbf{f}d\mathbf{y}$$

$$= \int_{\mathcal{R}^K} \psi(\mathbf{y})\delta(\mathbf{x}-\mathbf{y})\,d\mathbf{y} = \psi(\mathbf{x})$$

第二个等式可由公式(2.1.5)得到，第三个等式可由公式(2.1.4)得到。

逆 CSFT 表明任何信号$\psi(\mathbf{x})$可以表示为不同频率的复指数函数的线性组合。在特殊频率上的 CSFT 代表相应的复指数基函数的贡献[③]。

多数一维傅里叶变换的性质可以推广到 K 维的情况，包括线性特性、平移、卷积等。这里我们只给出卷积定理，它对信号分析是很重要的。

定理 2.2（卷积定理） 两个 K 维信号在空间域的卷积等价于这两个信号在频域的乘积，即：

$$\phi(\mathbf{x}) = \psi(\mathbf{x}) * h(\mathbf{x}) \leftrightarrow \Phi_c(\mathbf{f}) = \Psi_c(\mathbf{f})H_c(\mathbf{f}) \tag{2.1.8}$$

相反地，两个信号在空间域的乘积对应于它们在傅里叶域的卷积，即：

$$\phi(\mathbf{x}) = \psi(\mathbf{x})h(\mathbf{x}) \leftrightarrow \Phi_c(\mathbf{f}) = \Psi_c(\mathbf{f}) * H_c(\mathbf{f}) \tag{2.1.9}$$

定义 2.5 一个具有 K 维输入输出信号的系统一般描述为：

$$\phi(\mathbf{x}) = \mathcal{T}\{\psi(\mathbf{x})\},\ \mathbf{x}\in\mathcal{R}^K \tag{2.1.10}$$

令 $\phi_1(\mathbf{x})$和 $\phi_2(\mathbf{x})$分别代表相应于输入信号$\psi_1(\mathbf{x})$和$\psi_2(\mathbf{x})$的系统的输出信号。如果对于任意 α_1、$\alpha_2\in\mathcal{R}$有：

$$\mathcal{T}\{\alpha_1\psi_1(\mathbf{x}) + \alpha_2\psi_2(\mathbf{x})\} = \alpha_1\phi_1(\mathbf{x}) + \alpha_2\phi_2(\mathbf{x}) \tag{2.1.11}$$

则这个系统是线性的。

进一步，如果：

$$\mathcal{T}\{\psi(\mathbf{x}+\mathbf{x}_0)\} = \phi(\mathbf{x}+\mathbf{x}_0) \tag{2.1.12}$$

则该系统是移不变性的。

如果系统既是线性的又是移不变性的，则称其为线性移不变的（LSI）[④]。一个 LSI 系统对冲激

---

① 在一维情况下，CSFT 通常称为连续时间傅里叶变换或简称为傅里叶变换，因为典型的一维信号是时间函数。

② 在本书中，我们用频率变量 f 定义 CSFT，而不是用角频率 $\Omega=2\pi f$。在我们的定义中，不需要用 2π 归一化，因为 f 中的每一个变换基函数都具有单位模。

③ 公式(2.1.6)中定义的傅里叶积分对$\mathcal{R}^K$ 中所有函数都不收敛。但是如果我们允许傅里叶变换包含 δ 函数，那么傅里叶分析可用于各种函数。在本书中，不讨论连续和离散时间情况下傅里叶变换的各种收敛条件。有关傅里叶变换的更严格的定义参见文献[9,2]。

④ 在一维情况下，LSI 称为 LTI，当一维信号是时间函数时，意味线性时移不变。

信号 $\delta(\mathbf{x})$的输出信号$\mathcal{T}\{\delta(\mathbf{x})\}$称为这个系统的冲激响应,通常记为 $h(\mathbf{x})$。它的 CSFT 称为系统的频率响应,记为 $H(\mathbf{f})$。

定理 2.3 在一个输入和输出信号属于$\mathcal{R}^K$ 的 LSI 系统中,相应于任何输入信号$\psi(\mathbf{x})$的输出信号 $\phi(\mathbf{x})$可以表示为$\psi(\mathbf{x})$与系统冲激响应 $h(\mathbf{x})$的卷积,即:

$$\phi(\mathbf{x}) = \psi(\mathbf{x}) * h(\mathbf{x}) \tag{2.1.13}$$

在频域,它们的关系为:

$$\Phi_c(\mathbf{f}) = \Psi_c(\mathbf{f}) H_c(\mathbf{f}) \tag{2.1.14}$$

我们要强调的是上述结果只是线性移不变系统的结果。显然,系统的频率响应 $H(\mathbf{f})$描述输入信号将如何在不同的频率上被系统所改变。一个 LSI 系统完全是由它的冲激响应 $h(\mathbf{x})$和频率响应 $H(\mathbf{f})$决定的。

2.2 多维离散空间信号和系统

前一节考虑的是定义在 K 维连续空间$\mathcal{R}^K$ 中的信号。也有一些应用场合信号本身是离散的,也就是说信号只定义在离散的点集上。我们总可以用一个整数矢量来标志每个离散的点,这样信号就被定义在 K 维整数空间$\mathcal{Z}^K = \{[n_1, n_2, \cdots, n_K]^T \mid n_k \in \mathcal{Z}, k \in \mathcal{K}\}$上,这里$\mathcal{Z}$是所有整数的集合。一般地,我们把这样的信号称为 K 维离散信号或序列。离散信号的一个特殊情况是连续信号的采样形式;下一章我们将讨论采样信号的采样过程和性质。在这一节中,我们正式定义多维离散信号和系统,以及它们的离散空间傅里叶变换的特征。

定义 2.6 K 维离散空间信号$\psi(\mathbf{n})$是 K 维离散变量 $\mathbf{n} = [n_1, n_2, \cdots, n_K] \in \mathcal{Z}^K$ 的函数。这个函数可以取任何值,可以是实数或复数,若只取实数则称其为实函数。

定义 2.7 两个离散空间信号$\psi(\mathbf{n})$和 $h(\mathbf{n})$的离散卷积定义为:

$$\psi(\mathbf{n}) * h(\mathbf{n}) = \sum_{\mathbf{m} \in \mathcal{R}^K} \psi(\mathbf{n} - \mathbf{m}) h(\mathbf{m}) \tag{2.2.1}$$

定义 2.8 离散 δ 函数 $\delta(\mathbf{n})$定义为:

$$\delta(\mathbf{n}) = \begin{cases} 1, \mathbf{n} = \mathbf{0} \\ 0, \text{其他} \end{cases} \tag{2.2.2}$$

与连续函数一样,存在与离散 δ 函数有关的特殊性质。用求和代替公式(2.1.3~2.1.5)中的积分可以得到这些性质。

定义 2.9(离散空间傅里叶变换) K 维离散信号$\psi(\mathbf{n})$的离散空间傅里叶变换(DSFT)[①]定义为:

$$\Psi_d(\mathbf{f}) = \sum_{\mathbf{n} \in \mathcal{Z}^K} \psi(\mathbf{n}) \exp(-\mathrm{j} 2\pi \mathbf{f}^T \mathbf{n}) \tag{2.2.3}$$

这里 $\mathbf{f} \in \mathcal{R}^K$ 表示 K 维离散频率变量。

注意,由于公式(2.2.3)右边的复指数项,DSFT 在每一维上都是周期性的,并且以 1 为周期。在 K 维空间的基本周期是一个单位超立方体,记为$\mathcal{I}^K = \{\mathbf{f} \mid f_k \in (-\frac{1}{2}, \frac{1}{2}), k \in \mathcal{K}\}$。这个

① 在一维情况下,DSFT 最好称为离散时间傅里叶变换或 DTFT,因为一维信号典型地是离散时间标号的函数。

基本周期在所有整数点处重复。由于这个原因，只需在一个单位超立方体$\mathcal{I}^K$上规定信号的DSFT。在下一章(3.1节)中，我们将用阵列的概念正式定义K维信号的周期性。

定理2.4(逆离散空间傅里叶变换) 离散空间信号可以通过下式由其DSFT重建：

$$\psi(\mathbf{n}) = \int_{\mathcal{I}^K} \Psi_d(\mathbf{f})\exp(j2\pi\mathbf{f}^T\mathbf{n})\,d\mathbf{f} \tag{2.2.4}$$

根据下式，上式便很容易证明：

$$\int_{\mathcal{I}^K} \exp(j2\pi\mathbf{f}^T\mathbf{n})\,d\mathbf{f} = \delta(\mathbf{n}) \tag{2.2.5}$$

CSFT的多数性质可以移植到DSFT。特别地，卷积定理仍然适用，即：

$$\psi(\mathbf{n}) * h(\mathbf{n}) \leftrightarrow \Psi_d(\mathbf{f})H_d(\mathbf{f}) \tag{2.2.6}$$

类似于连续空间的情况，我们可以用离散输入和输出信号定义一个离散空间系统。如果这个系统是线性移不变的，那么它可以完全由冲激响应，即对于离散δ函数的系统输出$h(\mathbf{n}) = \mathcal{T}\{\delta(\mathbf{n})\}$来表征。对于任何输入信号的输出可以由这个输入信号与冲激响应的卷积来表示。

2.3 视频信号的频域特性

视频是一个三维信号，具有两个空间维(水平和垂直)和一个时间维。我们可以把上节描述的CSFT和DSFT直接以$K=3$分别用于模拟和数字视频信号。也可以把多维线性系统的各种概念和性质用于视频处理系统。代替用一般符号(x_1,x_2,x_3)描述一个三维信号域的点，我们将使用(x,y,t)，用x和y指示水平和垂直位置，用t指示时间维。类似地，我们将用(f_x, f_y, f_t)表示与这些坐标相联系的频率。读者自然会提出的问题是，这些频率的物理意义是什么以及视觉系统如何感知它们。在本节中，我们要回答第一个问题。在下一节中，将描述人类视觉系统的频率响应。

2.3.1 空间和时间频率

空间频率 二维空间频率是在二维图像平面上图像亮度或彩色变化快慢的一个度量。我们可以测量不同方向上的空间频率。在给定方向上的空间频率是用该方向上每单位长度的周期来测量的，单位长度可以是1米，也可以是电视监视器的像高等。二维图形的空间变化可以完全由两个垂直方向上的频率来表征。可以把任何方向上的频率投影到这两个方向上。通常，我们用一对值(f_x, f_y)表征二维图像信号的空间频率，它们分别表示水平和垂直频率。例如，用$\psi(x,y)=\sin(10\pi x)$表示的一个正弦图形的频率是(5,0)，其中在水平方向上每单位长度变化5个周期，而在垂直方向上保持不变。另一方面，图像$\psi(x,y)=\sin(10\pi x+20\pi y)$的频率为(5,10)，因为在水平和垂直方向上每单位长度分别有5和10个周期。这幅图像也可以用每单位长度周期为$\sqrt{f_x^2+f_y^2}\approx 11$且沿着$\theta=\arctan(f_y/f_x)\approx 64°$方向的一个频率来表示。这两个纯正弦图形如图2.1所示。

上面的例子是具有单一频率的纯正弦形式。利用傅里叶变换，任意信号都可以被分解成许多正弦形式，如公式(2.1.7)中的逆CSFT所描述的那样。

角频率 前面我们定义了根据给定单位长度在特定方向上的空间频率f_s。然而这种测量

不是很有用的,因为所感知的信号变化速度随着观察距离而增加。更有用的空间频率测量是根据每观测角度的周期数。如图 2.2 所示,如果图像高度为 h,观察距离为 d,当 $h/2 \ll d$ 时,垂直观测张角 θ 可由下式近似:

$$\theta = 2\arctan\left(\frac{h}{2d}\right)(\text{弧度}) \approx \frac{h}{d}(\text{弧度}) = \frac{180}{\pi}\frac{h}{d}(\text{度}) \tag{2.3.1}$$

如果每像高有 f_s 个周期,那么每观测度的周期数为:

$$f_\theta = f_s/\theta = \frac{\pi}{180}\frac{d}{h}f_s(\text{周期 / 度}) \tag{2.3.2}$$

我们称 f_θ 为角频率,它的单位是周期/度,或者 cpd。公式(2.3.2)表明了空间频率与角频率之间的关系。对于同样的图像,f_θ 随观测距离的增加而增加。另一方面,对于固定的观测距离,较大的屏幕尺寸导致较低的角频率。这些结果与我们的直觉匹配得很好:当在相当远处观看时,同样的图像显得变化得较快,而如果通过较大的屏幕观看,则变化得较慢。尽管上述关系是通过计算垂直方向的周期数决定的,角频率也可以沿水平方向或任何其他方向定义。注意,角频率并不只是信号本身的表征。它是由信号的空间频率和观测条件二者决定的。

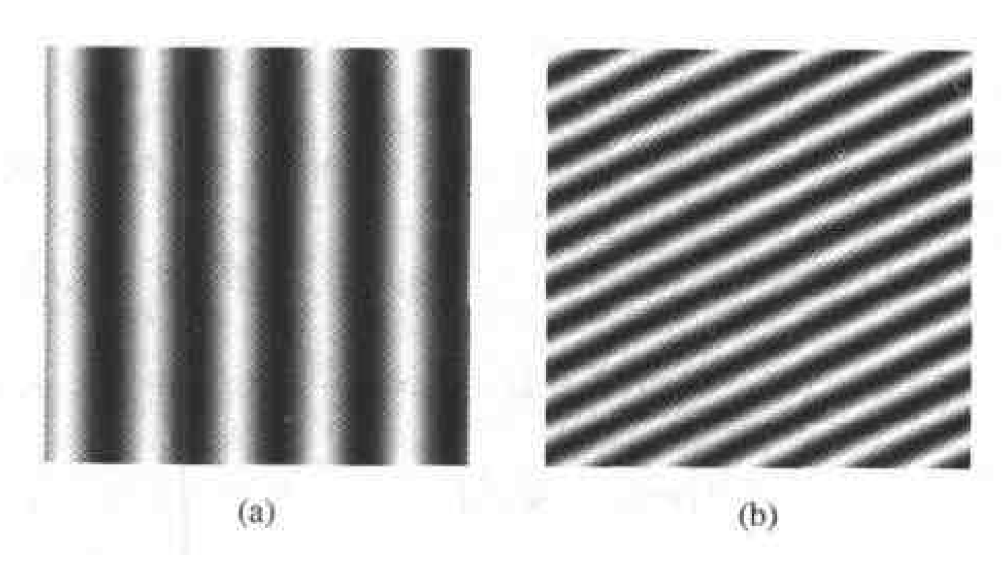


图 2.1 二维正弦信号:(a)$(f_x, f_y) = (5,0)$;(b)$(f_x, f_y) = (5,10)$。水平和垂直单位分别是图像宽和高。因此,$f_x = 5$意味着每行有5个周期

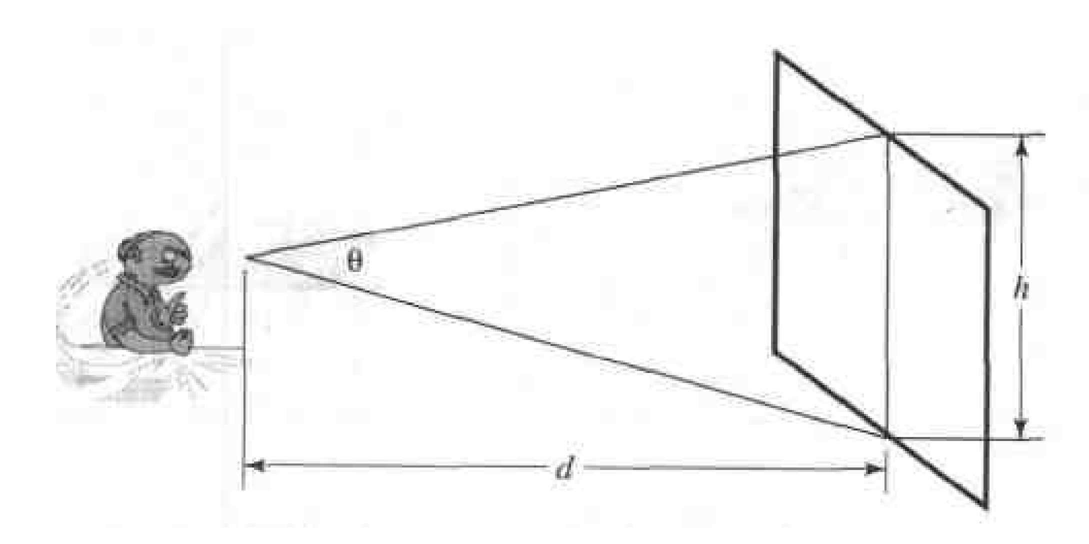


图 2.2 观测角与观测距离的关系;见公式(2.3.2)

时间频率 对于一维的时间函数,时间频率的定义是很清楚的:每秒周期数。对于由变化的二维帧组成的视频信号,时间频率是由二维位置决定的。对于一个固定的二维位置(x, y),它的时间频率定义为每秒周期数,通常记为 Hz。视频信号的最大时间频率是指在成像区域内所有点的时间频率的最大值。

2.3.2 由于线性运动而产生的时间频率

视频信号的时间频率取决于成像景物的变化速率，它可以是由摄像机运动、物体运动或二者的共同运动而引起的。通常在一个场景中有许多可能进行着不同运动的物体。因此，把运动与时间频率直接联系起来并不容易。这里，我们考虑与一个进行着线性(即以恒定速率)运动的物体相联系的时间频率。这个分析也用于当整个景物进行着例如由摄像机平移引起的总体线性运动的情况。正如我们将要看到的，时间频率不只是由运动决定的，也决定于物体的空间频率。

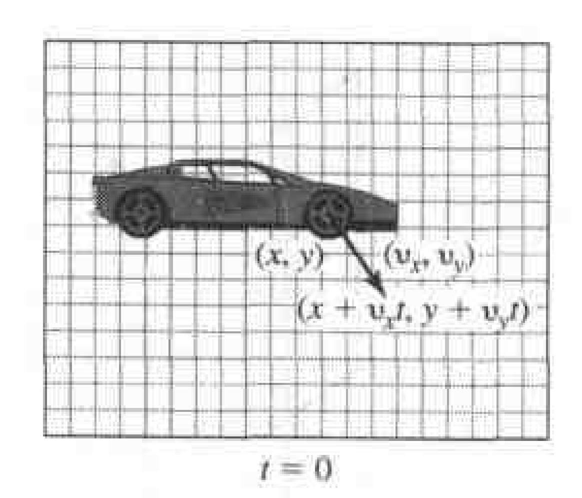
令时刻 0 物体的成像图形用$\psi_0(x,y)$表示，它在水平和垂直方向上的速度用 v_x 和 v_y 表示。设景物处于各向均匀的环境照度下，使得同一个物体点在不同时刻具有相同的图像亮度，则在时刻 t 的成像图形将是：

$$\psi(x,y,t) = \psi_0(x - v_x t, y - v_y t) \tag{2.3.3}$$

因为时刻 t 的点(x, y) 对应于时刻 0 的点$(x - v_x t, y - v_y t)$(见图 2.3)。这称为恒定亮度假设。这种假设有效的条件将在 5.2 节中进一步解释(公式(5.2.11))。对上述信号进行 CSFT，我们得到：

$$\begin{aligned}
\Psi(f_x, f_y, f_t) &= \iiint \psi(x,y,t)\exp(-j2\pi(f_x x + f_y y + f_t t))\,dxdydt \\
&= \iint \psi_0(x - v_x t, y - v_y t)\exp(-j2\pi(f_x(x - v_x t) \\
&\quad + f_y(y - v_y t)))\,dxdy \cdot \int \exp(-j2\pi(f_t + f_x v_x + f_y v_y)t)\,dt \\
&= \Psi_0(f_x, f_y)\int \exp(-j2\pi(f_t + f_x v_x + f_y v_y)t)\,dt \\
&= \Psi_0(f_x, f_y)\,\delta(f_t + f_x v_x + f_y v_y)
\end{aligned}$$

这里$\Psi_0(f_x, f_y)$表示$\psi_0(x,y)$的二维 CSFT。最后一个等式是基于公式(2.1.5)的。



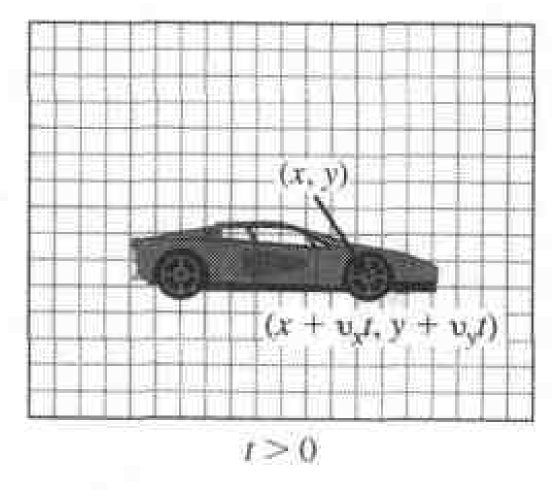


图 2.3 运动下的恒定强度假设的图示。在 $t=0$ 的每个点(x, y) 移动到时刻t时的点$(x + v_x t, y + v_y t)$，移动距离为$(v_x t, v_y t)$. 彩色或强度不改变。或者说，时刻t的点(x,y)对应于时刻0的点$(x - v_x t, y - v_y t)$

这个结果表明$\Psi(f_x, f_y, f_t)$只在由下述公式定义的平面上是非零的：

$$f_t + f_x v_x + f_y v_y = 0 \tag{2.3.4}$$

这意味着如果物体以速度(v_x, v_y)运动①，那么由物体的(f_x, f_y)表征的空间图案将导致一个时间频率：

$$f_t = -f_x v_x - f_y v_y \tag{2.3.5}$$

图 2.4(a)说明了对于给定的速度，f_x，f_y 以及f_t 之间的关系。我们可以看到，在三维空间中频谱的非零区域是一个由$(v_x, v_y, 1)$定义法向矢量的平面。设空间信号具有有限带宽$(f_{x,\max}, f_{y,\max})$，则时间带宽为 $f_{t,\max} = v_x f_{x,\max} + v_y f_{y,\max}$。在图 2.4(a)中，$f_{x,\max} = f_{y,\max} = 5$，因此，对于$(v_x, v_y) = (3,4)$，则有 $f_{t,\max} = 35$。

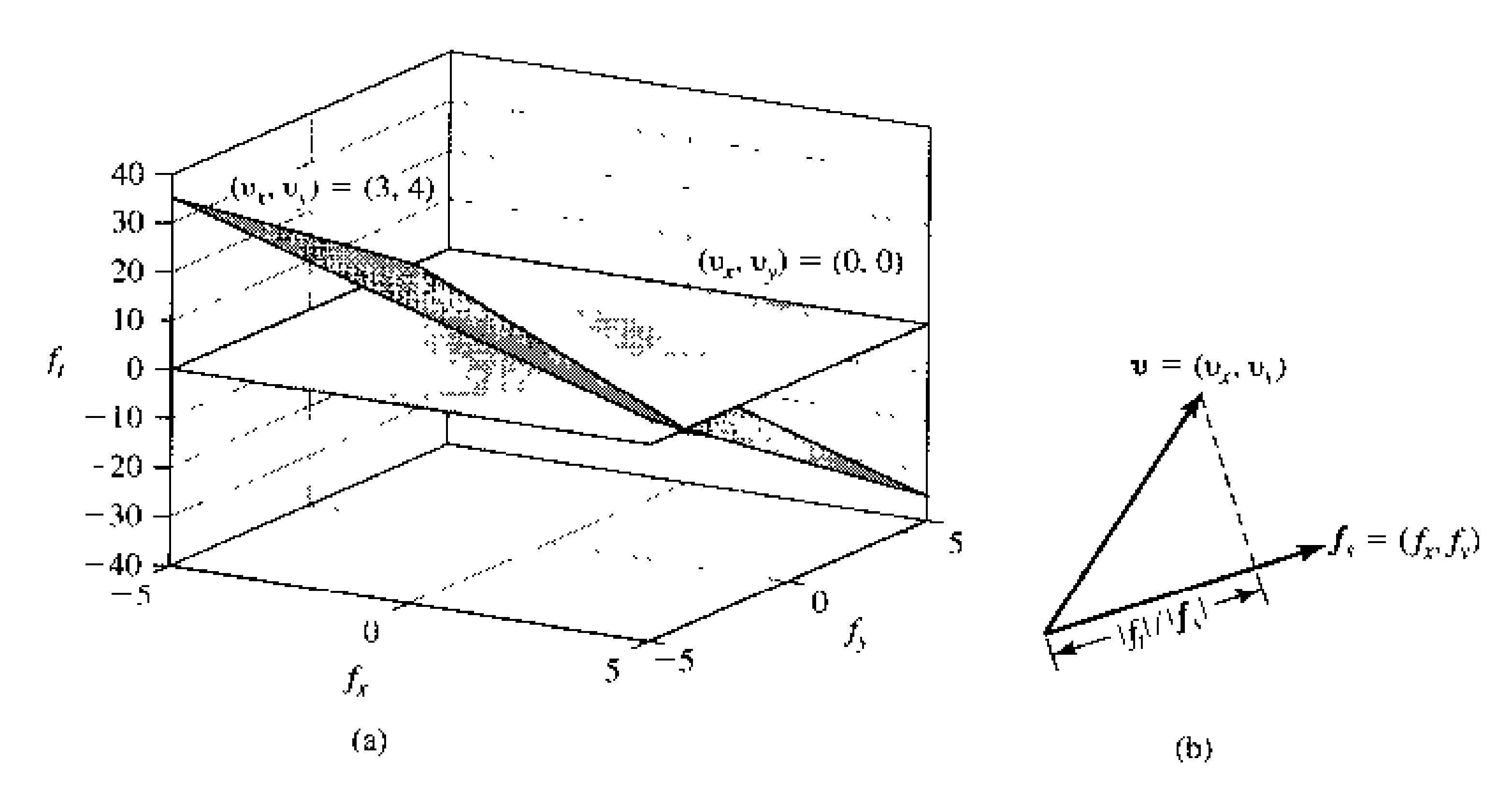


图 2.4　线性运动下空间与时间频率之间的关系。(a)相应于两个不同的速度矢量的(f_x, f_y, f_t)空间中的空时频率平面；(b)时间频率等于该速度在空间梯度上的投影

由公式(2.3.5)，时间频率不只是由速度决定的，也决定于空间频率。事实上，它是速度矢量在空间频率矢量上的投影，如图 2.4(b)所示。可以立即得到下面的结论：

- 当 $f_x = f_y = 0$ 时，不管 v_x, v_y 的值为多少，都有 $f_t = 0$。这意味着如果物体具有完全平坦的图形(即具有均匀的亮度或彩色)，那么无论物体沿着时间平面运动得多么快也不会观测到时间的变化。
- 如果运动方向(v_x, v_y)与空间频率方向(f_x, f_y)是正交的，那么时间频率 $f_t = 0$。空间频率的方向(f_x, f_y)是变化速度最高的方向。它的垂直方向是没有空间变化的方向。这个结果表明物体在其图形不变的方向上运动不会产生任何时间变化。当物体在空间变化最大的方向上运动时，时间频率最大。

考虑一个具有正弦竖条图案的有限平面，如图 2.1(a)所示。在这种情况下空间频率的方向是水平的。如果平面垂直地从顶端移动到底端(垂直于空间频率的方向)，那么人眼是感觉不出任何变化的，不管平面运动得多快。一旦这种运动稍微偏离垂直方向，人眼就会开始感觉到时间的变化。当平面水平地从左向右移动时(沿着空间频率的方向)，感觉到的变化最快。

① 注意，对于实信号 CSFT 是对称的，因此对于每一个频率分量(f_x, f_y)，也存在一个具有同样幅度的分量$(-f_x, -f_y)$。由这个分量产生的相应的时间频率是$f_x v_x + f_y v_y$。

上述分析是对于进行恒定速度运动的物体得到的。对于包含较复杂景物的视频，多个物体以不同方式运动，我们可以把图像区域分成小的区域，以便使每个小区域可以认为是以恒定的速度进行同样的运动。在这个区域的局部空间频率和速度决定了与这个区域有关的局部时间频率。

2.4 人类视觉系统的频率响应

视频系统最终是以观看者为对象的。因此，理解人类视觉系统如何感觉视频信号是极其重要的。在1.1节中，我们描述了彩色感觉机制；在本节中，我们集中于对图像亮度的时间和空间变化的感觉。正如将要展示的，HVS对视觉图形的敏感度是由图形的时间和空间频率成分决定的。在某些中间的空间和时间频率上，视觉敏感度是最高的。然后敏感度迅速下降，并在某些截止频率上退隐。这些频率以上的空间和时间变化对于人眼是不可见的。视觉频率响应的知识在设计视频系统中是非常重要的。例如，时间和空间截止频率是确定视频摄取和显示系统的帧率和行率的基础。下面，我们首先描述HVS的空间和时间频率响应，然后介绍联合的空时频率响应，最后，我们描述当人眼跟踪运动物体时如何转换视觉频率响应。

2.4.1 时间频率响应和闪烁特性

HVS的时间频率响应是指对不同频率的时间变化图形的视觉敏感度。已经进行了很多实验来确定HVS的时间频率响应。研究结果表明观察者的时间响应决定于许多因素，包括观测距离、显示亮度和环境照明。图2.5示出了由Kelly[5]所做的实验得到的结果。在这个实验中，为观测者提供了一个平面屏幕，其亮度以下面的形式成正弦变化：

$$\psi(t) = B(1 + m\cos 2\pi ft) \tag{2.4.1}$$

对于固定的平均亮度级 B 和频率 f，调制度 m 是变化的，要求观察者确定最低调制度 m_{min}，此时屏幕亮度的时间变化（即闪烁）刚刚变得不可觉察。显然，m_{min} 的倒数表示观察者对于给定频率 f 的时间变化的敏感度。因此，$1/m_{min}$ 也称为对比敏感度，用于描述视觉敏感度或视觉响应。对比敏感度作为频率的函数称为调制传输函数（MTF）。

可以看到，HVS的时间响应类似于一个带通滤波器的响应，在某个中间频率点达到峰值，然后响应迅速下降，直到大约4.5倍的峰值频率。峰值随着图像的平均亮度增加。例如，当平均亮度为0.65筹阑时①，峰值响应出现在大约5 Hz，而截止频率，也就是响应足够小的地方，大约是20～25 Hz。另一方面，当平均亮度为850筹阑时，响应在大约15～20 Hz时最高，大约在75 Hz时退隐。

人眼在较高时间频率处敏感度降低的一个原因是人眼对图像的感觉能够保持一段短的时间，即使这时实际图像已经移走了。这种现象称为视觉暂留②。一方面，当图形变化速率比HVS的刷新率更快时，它会引起被观察图形时间上的模糊。另一方面，视觉暂留使视频信号显

① 筹阑（troland）是用于描述进入视网膜的光的强度的一种单位。

② 这是由HVS的时间相加机制引起的，它对进入的光进行积分。布劳赫（Bloch）定律指出，积分或维持时间与光的强度成反比[10]。光源越亮，积分时间越短。当显示较亮时，它给予人眼以较高的时间敏感度，这与图2.5所示的结果是一致的。

示为一个连续的帧序列。只要帧间隔比视觉暂留周期短，则人眼就会感觉到一个连续变化的图像。当帧间隔比视觉暂留周期长时，人眼就会观察到帧闪烁，这种现象是指不连续帧的感觉。人眼察觉不到闪烁的最低帧率称为临界闪烁频率，它与前面讨论的时间截止频率在定义上是等价的。视频摄取或显示系统采用的帧率应该大于临界闪烁频率以避免闪烁感觉。

从图2.5中可以看出，临界闪烁频率的范围是从20到80 Hz，取决于显示的平均亮度。显示得越亮，临界频率就越高。在电影院里，平均亮度是很低的；另一方面，CRT显示就要亮得多，接近9 600 筹阑[8]。这就是为什么动画系统可以采用低于电视系统的帧率。目前，动画业采用24帧/秒(fps)，而电视业采用50(在PAL和SECAM系统中)和60场/秒(在NTSC系统中)。我们可以看到，这些帧率和场率都接近于与相关应用场合的平均亮度水平相联系的截止频率。计算机显示采用高得多的帧率72 fps，因为我离计算机屏幕比离电视屏幕近得多，而在较短的距离视觉阈值较高。3.3.1节给出了关于如何确定视频系统的空间-时间采样率的更详细的讨论。

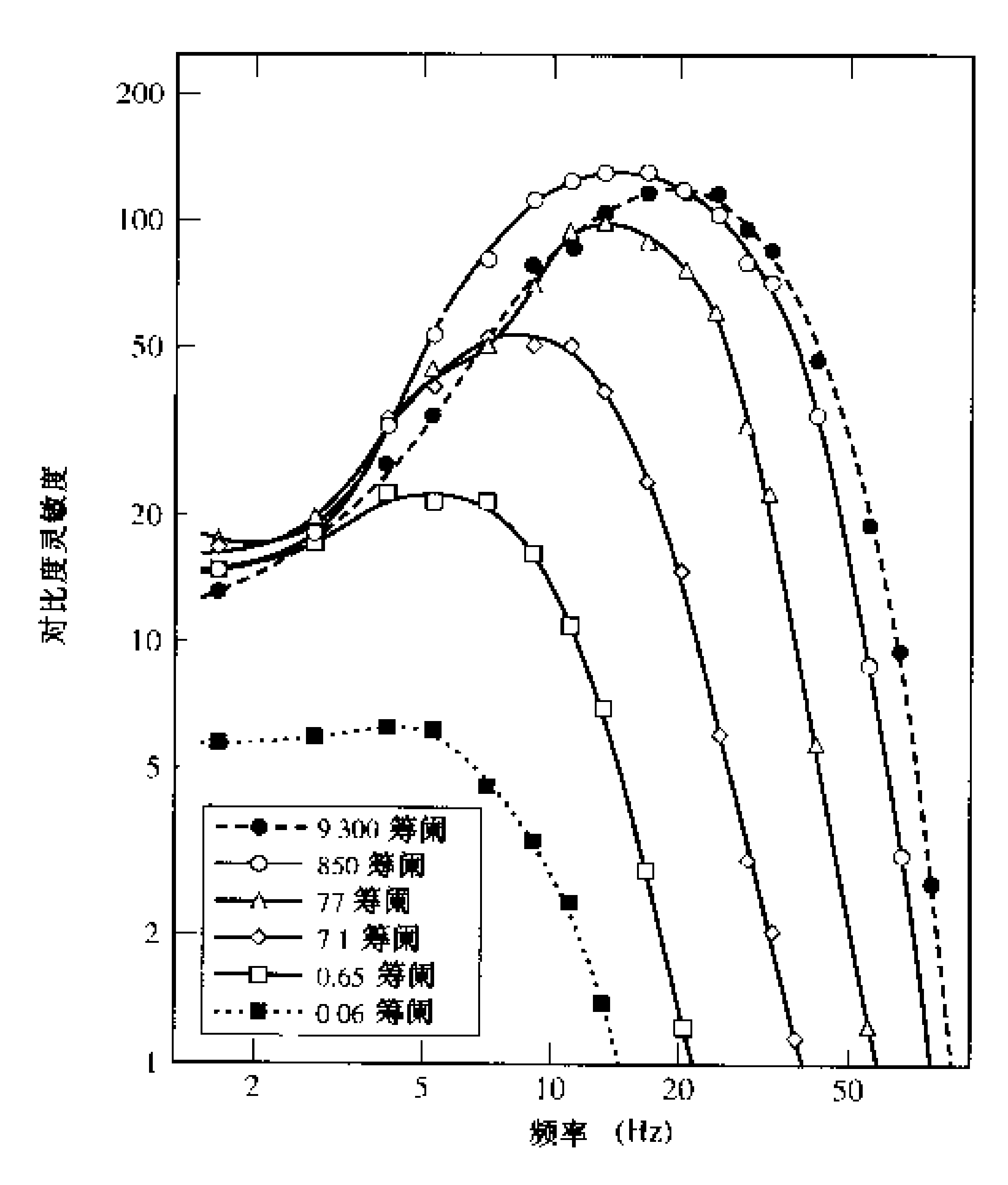


图2.5　由一个视觉实验获得的HVS的时间频率响应。不同的曲线代表在不同的亮度水平B(以筹阑为单位)下得到的响应。水平轴代表闪烁频率f，单位为Hz

2.4.2　空间频率响应

HVS的空间频率响应是指对于具有不同空间频率的静止空间图形的视觉敏感度。假设视觉敏感度关于空间变化的方向是各向同性的，那么就可以对于一个任意的空间方向计算出空

间频率响应,通常是用水平和垂直方向[①]。为了使响应独立于观测的距离,空间频率响应通常表示为角频率的函数。为确定 HVS 的空间频率响应已经做了很多实验。结果发现 HVS 的空间频率响应也类似于一个带通滤波器,峰值响应大约在 2~5 周/度(cpd),在大约 30 cpd 处退隐。由 Kelly[6]进行的研究结果如图 2.6 所示。这个结果是通过观察在水平方向上正弦变化的一个竖条图形得到的,即:

$$\psi(x,y,t) = B(1 + m\cos 2\pi fx) \tag{2.4.2}$$

这个图形显示在具有很高刷新率(1 000 fps)的 CRT 监示器上,因此可以认为它在时间上是静止的。对每个给定的空间频率 f,调制度 m 是变化的,要求观察者确定最小的调制度 m_{min},此时空间变化刚刚变得不可觉察。图 2.6 的垂直轴表示 $1/m_{min}$。图中示出了三条曲线,它们是借助为消除人眼运动影响所设置的各种稳定器得到的。

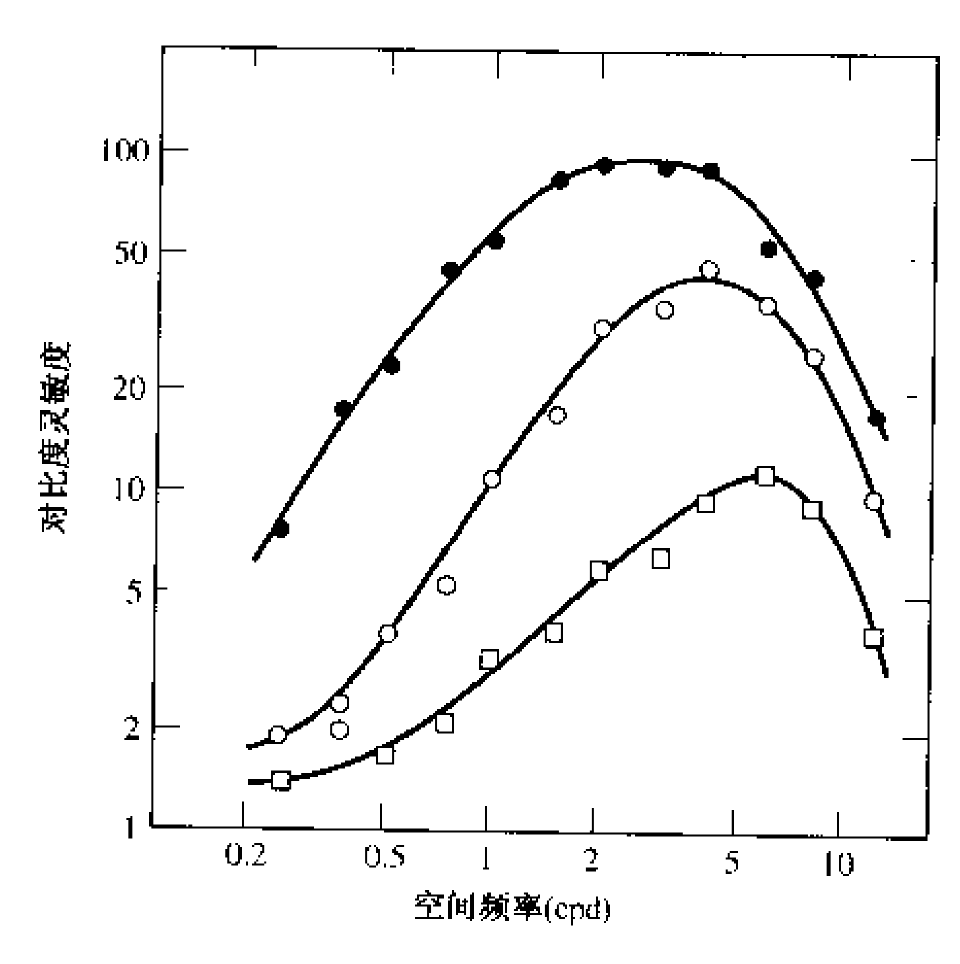


图 2.6 由一个视觉实验获得的 HVS 的空间频率响应。三条曲线是由不同的为消除扫视眼运动影响的稳定器得到的。实心圆是在正常的没有稳定器的条件下获得的;空心方块是在具有最佳增益的稳定器下获得的;空心圆是在增益大约变化5%下获得的

当观看景物时,眼睛经常快速地从一个固定位置跳到另一个位置。这称为扫视眼运动[4]。研究已经发现,扫视眼运动有增强对比度敏感度的作用,但是会降低峰值响应出现的频率。例如,在图 2.6 中,正常眼运动比没有眼运动(即戴着稳定器)的观察者敏感度大约高 10 倍。另一方面,当具有正常眼运动时,峰值响应出现在大约 2 cpd,但是当完全去除眼运动时,峰值响应大约移到 4 cpd。

空间频率响应的带通特性可以归因于 HVS 有效进行的带通滤波工作。已经发现 HVS 的冲激响应波形类似于高斯函数的拉普拉斯运算(墨西哥草帽形状),在视觉细胞附近是正的权重,而对于较远的细胞是负的权重。正的部分起平滑作用,而负的部分有效地增强尖锐边缘。负加权效果是由视觉皮层的侧抑制引起的[1]。

① 实际上,视觉敏感度对于水平和垂直方向上的变化比其他方向高。

在 3.3.1 节，我们将说明在确定视觉系统的水平和垂直采样频率时如何考虑 HVS 的空间频率响应。

2.4.3 空时频率响应

前面给出的空间频率响应是在零时间频率定值的，反过来也一样。这里我们讨论空间和时间频率都是非零时的视觉响应。已经发现，在较高的时间频率上，空间频率响应的峰值频率和截止频率都向下偏移。对于时间频率响应也有类似的倾向。

图 2.7 示出了 Robson[12]的实验结果。这个实验中的测试图形由下式表述：

$$\psi(x,y,t) = B(1 + m\cos(2\pi f_x x)\cos(2\pi f_t t)) \tag{2.4.3}$$

对于一个 f_x 和 f_t 的固定组合，调制度 m 是变化的，要求观察者确定最小的调制度 m_{min}，此时空间和时间的变化刚刚不可觉察。图 2.7 中的垂直轴代表 $1/m_{min}$。我们可以看到，在零时间（空间）频率附近，空间（时间）频率响应具有带通特性。这与前面所示的结果是一致的。但是在较高的时间（空间）频率上，空间（时间）频率响应变得更像一个低通滤波器，当时间（空间）频率增加时峰值响应下降。这表明当成像图形运动得很快时，人眼不能区分图形中很高的空间频率。当图像静止时，人眼能够分辨高得多的空间频率。类似地，在较高的空间频率上，时间响应变成低通，并且在较高的空间频率上过渡频率向下偏移。

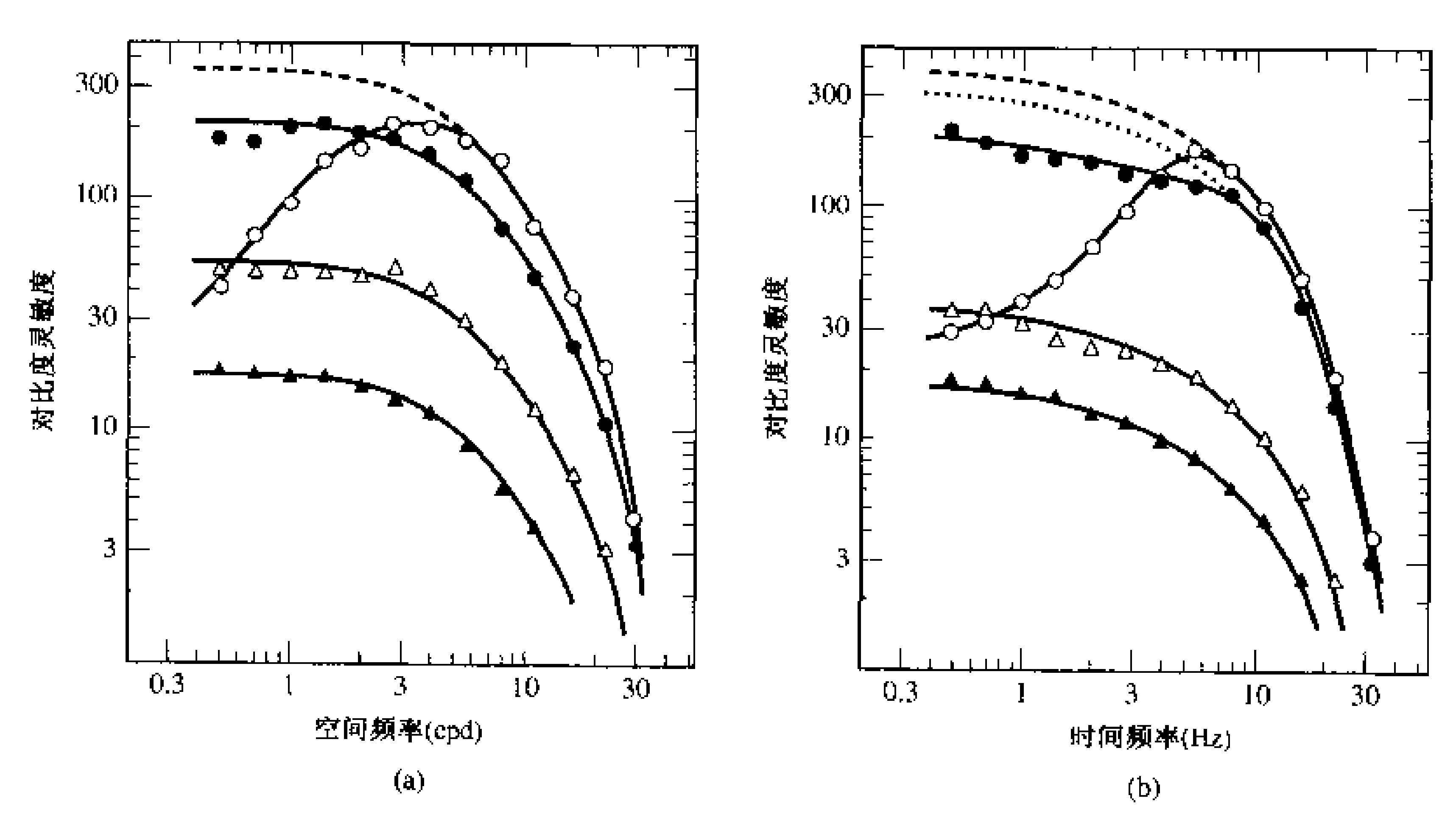


图 2.7　HVS 的空时频率响应。(a)对于不同的时间频率 1 Hz(空心圆)、6 Hz(实心圆)、16 Hz(空三角)和22 Hz(实三角)的空间频率响应。(b)对于不同的空间频率0.5 cpd(空心圆)、4 cpd(实心圆)、16 cpd(空三角)和22 cpd(实三角)的空间频率响应

这个空间与时间频率响应之间的彼此相反的关系，对于视频系统的设计意味着可以用时间分辨率换取空间分辨率，反过来也一样。这个性质已经被明智地利用于模拟电视系统的隔行扫描机制中。对于隔行扫描，为了用有限的帧率呈现快速变化的场景，一帧被分成两场，每场的行数是一帧的一半。因为当时间频率高的时候，HVS 具有降低的空间频率响应，所以人眼不能区分快速变化场景中很高的空间频率，即使它是以很高的空间分辨率呈现的。因此，采用

较低的空间分辨率是合适的。另一方面，当成像场景静止时，在两个分离场中的行组合起来提供一个高的空间分辨率，以适应 HVS 的较高的空间频率分解能力。

空间与时间响应之间的彼此相反性也解释了为什么当我们近距离观看电视时会感觉到行闪烁。在这种情况下，行被进一步分开而角频率变低。在低的角频率下，人眼在时间上更敏锐，因此较容易感觉到闪烁。

Robson 的实验是在正常扫视眼运动下进行的，后来 Kelly 又进行了一个实验，用稳定技术消除扫视运动的影响[7]。为测试时间响应，他用行波代替闪烁波。得到的空时频率响应的趋势是类似的，但是由运动引起的时间响应大约是由闪烁波引起的两倍。这意味着人眼对由运动引起的时间变化比由闪烁引起的时间变化更敏感。

2.4.4 平滑追踪眼运动

尽管 2.4.1 节所述的实验表明 HVS 不能分辨大于 80 Hz 的时间频率，但实际上当人眼追踪物体时，例如当棒球击球手追踪快球时，人眼能够看清快速运动的物体。这是因为当人眼追踪物体时，物体的相对运动变得较小。如果达到完全追踪，那么物体看起来是静止的。人眼自动运动以追踪被观察物体的现象称为平滑追踪眼运动[11,4]。

为了说明人眼的追踪效应，我们需要把显示屏幕上的时空频率转移到人眼移动时的视网膜坐标[3]。设人眼是以速度 v_x 和 v_y 运动，那么在视网膜坐标下观测的图像 $\tilde{\psi}(x,y,t)$ 与显示屏幕上的图像 $\psi(x,y,t)$ 之间的关系为：

$$\tilde{\psi}(x,y,t)=\psi(x+v_xt,y+v_yt,t) \tag{2.4.4}$$

两边进行 CSFT 得：

$$\tilde{\Psi}(f_x,f_y,f_t)=\Psi(f_x,f_y,f_t-v_xf_x-v_yf_y) \tag{2.4.5}$$

这种关系意味着视网膜上的观测频率 $(\tilde{f}_x,\tilde{f}_y,\tilde{f}_t)$ 与显示屏幕上的频率 (f_x,f_y,f_t) 之间的关系为：

$$\tilde{f}_x=f_x,\ \tilde{f}_y=f_y,\ \tilde{f}_t=f_t+v_xf_x+v_yf_y \tag{2.4.6}$$

当被显示物体的时间频率为零时，感觉到的时间频率随着眼睛运动速度和被观测物体空间频率的增加而增加。当被观测物体以恒定速度运动时，产生的时间频率由公式(2.3.5)所描述。因此，当眼运动与物体运动匹配时，视网膜上的有效时间频率为零。一般地，通过追踪物体运动，视网膜上的有效时间频率会降低。

利用公式(2.4.6)的坐标变换，Griod 重画了 Robson 测量的视觉频率响应。图 2.8 示出了没有平滑追踪眼运动以及在两个不同的眼运动速度下的视觉频率响应。在这个图中，对比敏感度的测量与图 2.5～图 2.7 不同，因此具有不同的尺度。显然，平滑追踪眼运动的效果是把视觉响应的非零区域扩展到一个较大的时间频率范围。事实上，可以察觉到高达 1 000 Hz 的时间频率[3,13]。

正如 Girod 所指出的[3]，平滑追踪眼运动对于视频显示和处理系统的设计是重要的。为了避免快速运动物体的可视模糊，显示系统必须能够显示达到 1 000 Hz 的时间频率。同样，进行任何类型的时间滤波必须带有运动补偿(即沿运动轨迹滤波)，以避免图像内容的可视模糊。

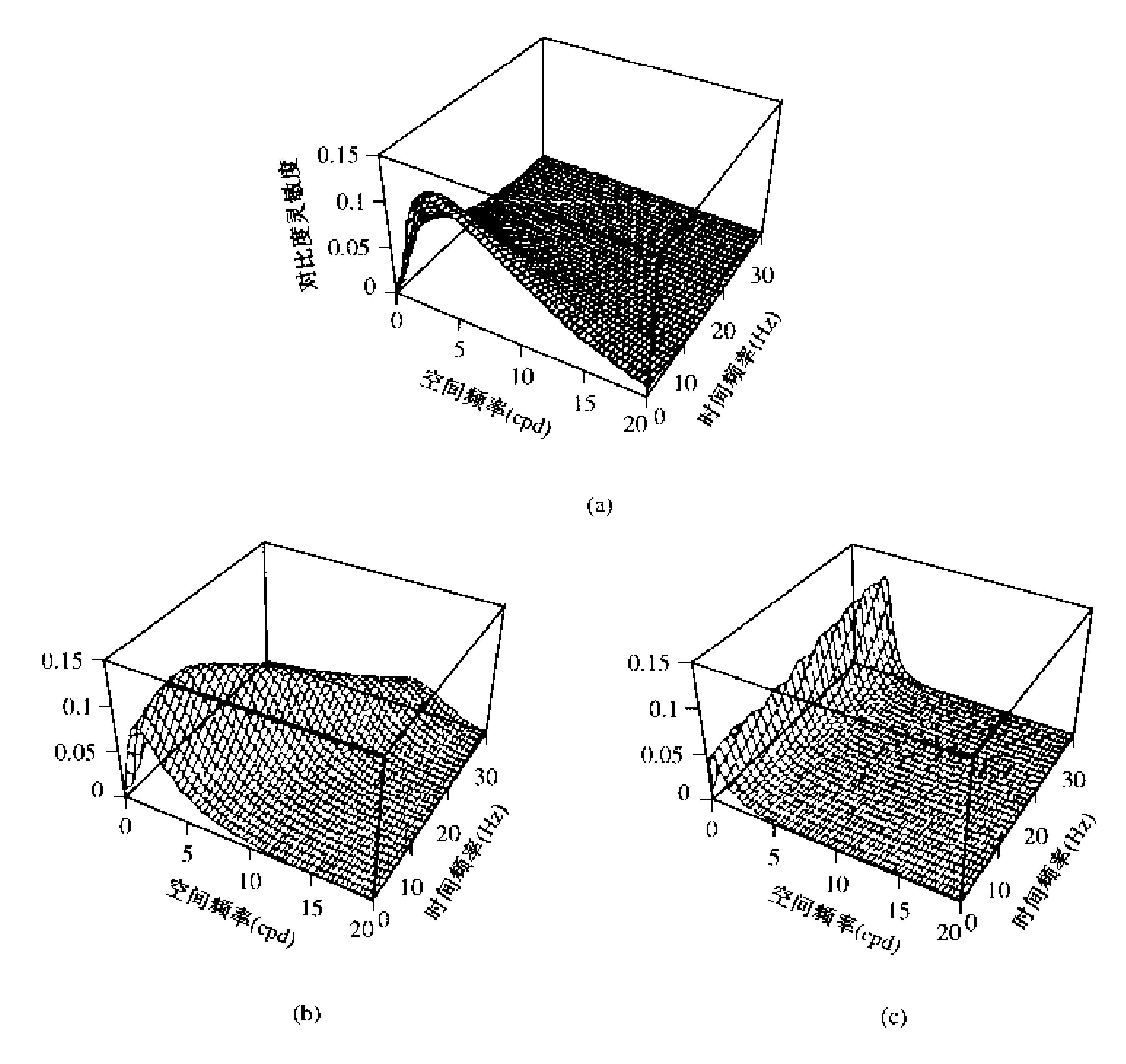


图 2.8　在平滑追踪眼运动下的 HVS 的空时响应:(a)没有平滑追踪眼运动;(b)在2度/秒的眼运动速度下;(c)在10度/秒的眼运动速度下

2.5　小结

多维信号与系统的傅里叶分析(2.1 及 2.2 节)

- 一个 K 维连续(离散)信号可以由一个连续(离散)空间傅里叶变换表示(公式(2.1.6 及 2.1.7)用于 CSFT,公式(2.2.3 及 2.2.4)用于 DSFT)。逆变换的存在意味着该信号可以分解为无限多个具有不同频率的复数正弦函数。
- 一个 LSI 系统可以完全由它的冲激响应,或者等价地由它的频率响应(即冲激响应的傅里叶变换)来表征(定理 2.3)。
- 一个 LSI 系统的输入和输出信号通过在空间域与冲激响应的卷积,或者通过在变换域与频率响应的乘积联系在一起,分别由公式(2.1.13)和(2.1.14)给出。

视频信号中的空间和时间频率(2.3 节)

- 视频信号的二维空间频率可以沿着任何两个正交的方向规定,通常沿水平和垂直方向。

- 运动物体的时间频率是由它的速度和空间频率决定的。更明确地说，任何图像点的时间频率与速度矢量在物体上该点的空间频率方向的投影成正比，如公式(2.3.5)和图2.4所示。

HVS 的空间和时间频率响应(2.4 节)

- 人眼对某个中间的空间和时间频率分量最敏感。在正常的观测条件下，人眼对2～5 cpd之间的空间频率及5～20 Hz之间的时间频率最敏感。
- 超过某些截止频率，眼睛不能觉察空间的变化。这些截止频率出现在空间频率大约为30 cpd，以及时间频率20～80 Hz之间。这些视觉阈值的组合为设计视频处理和显示系统提供指导，系统只需提供两倍于视觉阈值的频率。
- 只要帧率(对于隔行显示是场率)超过临界闪烁频率，对于大多数人来说是80 Hz以下，人眼将感觉不到一个个的帧。
- 视觉频率响应取决于显示的平均亮度。一般情况下，较亮的显示可以使人眼对高频分量更敏感。
- 当人眼追踪运动物体时，在视网膜上感觉的时间频率降低。当完全追踪时，视网膜平面上的时间频率为零。
- 在视频摄取、处理和显示中，HVS 的截止频率是确定空间和时间采样分辨率的主导因素(这个问题将在3.3.1节中进一步讨论)。

2.6 习题

2.1 摄像机的冲激响应通常模型化为：

$$h(x,y,t)=\begin{cases}\dfrac{1}{T_xT_yT_e} & |x|<\dfrac{T_x}{2},|y|<\dfrac{T_y}{2},t\in(0,T_e)\\ 0 & \text{其他}\end{cases}\tag{2.6.1}$$

这里 T_x，T_y 是摄像机光圈的水平和垂直尺寸，T_e 是曝光时间。求出 $h(x,y,t)$ 的CSFT。画出幅度响应。并讨论参数 T_x，T_y，T_e 对摄像机频率响应的作用。

2.2 假设问题2.1所述的摄像机正在摄取一个屏幕，其中含有一个各边宽度为 B 并与摄像机成像平面平行运动的立方体。投影在摄像机平面上的图像可表示为：

$$\psi(x,y,t)=\begin{cases}1 & -B/2+v_xt<x<B/2+v_xt,\ -B/2<y<B/2\\ 0 & \text{其他}\end{cases}\tag{2.6.2}$$

求出摄像机所摄取的信号。假设 $B\gg T_x$，$B\gg T_y$。

2.3 当设计一个视频摄取或显示系统时，为确定帧率和行数你会有哪些考虑？选择的这些参数适用于NTSC彩色电视广播吗？

2.4 为什么计算机监视器比典型的电视机监视器采用更高的时间刷新率和行数？

2.5 考虑电视屏幕上的一个具有100周期/像高的横条图形。如果像高为1米，观众坐在离屏幕3米远的地方，那么以cpd为单位的等效角频率是多少？如果观众坐在1米或5米远的地方呢？在每种情况下，观众能够恰当地感觉到垂直变化吗？

2.6 考虑一个平坦的、最大空间频率为$(f_x, f_y) = (3,4)$周/米的各向均匀的纹理表面,它以恒定的速度$(v_x, v_y) = (1,1)$米/秒运动。物体表面任何一点的时间频率是多少?如果速度为以下各值(单位是米/秒):(4,-3),(4,0),(0,1),结果又怎样呢?

2.7 (接2.6题)假设人眼以与物体相等的速度追踪运动的物体。对于不同的速度,视网膜上感觉到的时间频率是多少?如果人眼以固定的速度(2,2)米/秒运动呢?

2.7 文献目录

[1] Cornsweet, T. N. *Visual Perception*. New York: Academic Press, 1970.

[2] Dudgeon, D. E., and R. M. Mersereau. *Multidimensional Digital Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1984.

[3] Girod, B. Motion compensation: visual aspects, accuracy, and fundamental limits. In Sezan, M. I., and R. L. Lagendijk, eds., *Motion Analysis and Image Sequence Processing*, Boston: Kluwer Academic Publishers, 1993, 126–52.

[4] Hallett, P. Eye movements. In Boff, K., L. Kaufman, and J. P. Thomas, eds., *Handbook of Perception and Human Performance: Sensory Processes and Perception*. New York: Wiley-Interscience, 1986.

[5] Kelly, D. H. Visual responses to time-dependent stimuli. I. Amplitude sensitivity measurements. *J. Opt. Soc. Am.* (1961), 51:422–29.

[6] Kelly, D. H. Motion and vision. I. Stabilized images of stationary gratings. *J. Opt. Soc. Am.* (1979), 69:1266–74.

[7] Kelly, D. H. Motion and vision. II. Stabilized spatio-temporal threshold surface. *J. Opt. Soc. Am.* (1979), 69:1340–49.

[8] Netravali, A. N., and B. G. Haskell. *Digital Pictures—Representation, Compression and Standards*. 2nd ed. New York: Plenum Press, 1995.

[9] Oppenheim, A. V., and R. W. Schafer. *Discrete-Time Signal Processing*. Englewood Cliffs: Prentice Hall, 1989.

[10] Owen, W. G. Spatio-temporal integration in the human peripheral retina. *Vision Research* (1972), 12:1011–26.

[11] Robinson, D. A. The mechanics of human smooth pursuit eye movement. *J. Physiology* (Oct. 1965), 180(3):569–91.

[12] Robson, J. G. Spatial and temporal contrast sensitivity functions of the visual system. *J. Opt. Soc. Am.* (Aug. 1966), 56:1141–42.

[13] Tonge, G. J. Time-sampled motion portrayal. In *Second International Conf. Image Processing and Its Applications*, 1986, London.

第3章 视频采样

任何数字视频处理工作的第一步都是把本质上连续的视频信号转换为数字视频信号。数字化过程包括两步:采样和量化。这可以用数字摄像机实现,它直接把连续的物理场景视频数字化,或者通过把模拟摄像机产生的模拟信号数字化来实现。我们也经常需要把数字视频信号从一种格式(在空间和时间分辨率方面)转化为另一种格式,例如把一个视频录像从 PAL 格式转化为 NTSC 格式。这一章我们考虑采样问题,采样率转换问题将在第 4 章讨论;量化问题将在第 8 章与信号压缩的其他方法一起讨论。

设计一个视频采样系统,必须解决三个问题:(1)在空间和时间方向上必需的采样频率是什么?(2)给定总采样率(即水平、垂直和时间采样率的乘积)时,怎样在三维空间采样才能得到最好的表示?(3)假设只能使用有限的采样率,如何避免采样的混叠效应?本章从介绍多维信号采样的一般理论入手,把信号频谱与它所需要的采样图案联系起来(3.1 节及 3.2 节)。然后重点讨论视频信号的采样(3.3 节)。首先描述决定视频信号所需采样率的因素,以及在实际系统中经常做出的权衡考虑。然后讨论二维采样方案并比较逐行和隔行扫描。我们也考虑如何采样彩色视频信号,并且作为一个例子再次回到 BT.601 格式。随后我们介绍几种三维采样点阵并比较它们的效率。最后,描述通用视频摄像机和显示器是如何隐含地完成预滤波和内插功能的,并讨论一些实际的局限性(3.4 节)。

3.1 点阵理论基础

在本节和下一节,我们把众所周知的一维和二维信号采样理论推广到一般的多维空间。大家回忆,对于一维信号,样点通常是以规则的间距抽取的。对于二维信号,样点通常是在矩形格栅上抽取的。实际上,也可以在非矩形格栅上抽取样点,只要这个格栅具有能用整数矢量来指定格栅点的结构。数学上把这种类型的格栅称为点阵。本节引入 K 维空域点阵的概念。下一节将讲述如何用点阵对连续信号进行采样。

在点阵上采样多维信号的理论首先是由 Petersen 和 Middleton[7]提出的。Dubois 出色地考察了相关的理论,并扩展到用在点阵中所选择的子点阵的陪集集合进行采样[4]。本章只介绍对于分析多维空间采样过程很重要的那些概念和性质。关于点阵理论详细的阐述,读者可参阅参考文献[2]。这里介绍的定义和定理主要基于文献[4]。

定义 3.1 实 K 维空间 $\mathcal{R}^K$ 中的点阵 Λ,是可表示为 K 个线性独立的基矢量 $\mathbf{v}_k \in \mathcal{R}^K, k \in \mathcal{K} = \{1,2,\cdots,K\}$ 的整数加权组合的所有可能矢量的集合。即:

$$\Lambda = \left\{ \mathbf{x} \in \mathcal{R}^K \middle| \mathbf{x} = \sum_{k=1}^{K} n_k \mathbf{v}_k, \forall n_k \in \mathcal{Z} \right\} \tag{3.1.1}$$

矩阵 $[\mathbf{V}] = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_K]$ 称为生成矩阵[①]。

① 它也称为采样矩阵。

例 3.1 考虑$\mathcal{R}^2$中具有以下生成矩阵的两个点阵：

$$[\mathbf{V}_1]=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix};[\mathbf{V}_2]=\begin{bmatrix}\sqrt{3}/2 & 0\\ 1/2 & 1\end{bmatrix} \tag{3.1.2}$$

为了根据给定的生成矩阵画出实际的点阵，我们首先画出对应于基矢量的两个点，然后确定出对应于基矢量典型的整数加权组合的点；例如 $\mathbf{v}_1+\mathbf{v}_2$，$\mathbf{v}_1-\mathbf{v}_2$ 等。基于这些点，我们通常可以通过视觉观察外推出其他所有可能的点。按照这个步骤，我们在图 3.1(a)，(b)中画出了根据两个给定的生成矩阵所确定的点阵。正如所预期的，第一个点阵为简单的矩形格栅，因为$[\mathbf{V}_1]$是对角矩阵。第二个点阵是所谓的六角形点阵。尽管一组基矢量或一个生成矩阵能够完全定义一个点阵，与一个点阵有关的基矢量或生成矩阵却不是惟一的。事实上，可以容易地找到生成相同点阵的一个以上的基矢量或生成矩阵。例如，可以证明，对于图 3.1(b)中的点阵，另外一组基矢量是 $\mathbf{v}_1=[\sqrt{3}/2,1/2]^T$，$\mathbf{v}_2=[\sqrt{3},0]^T$。

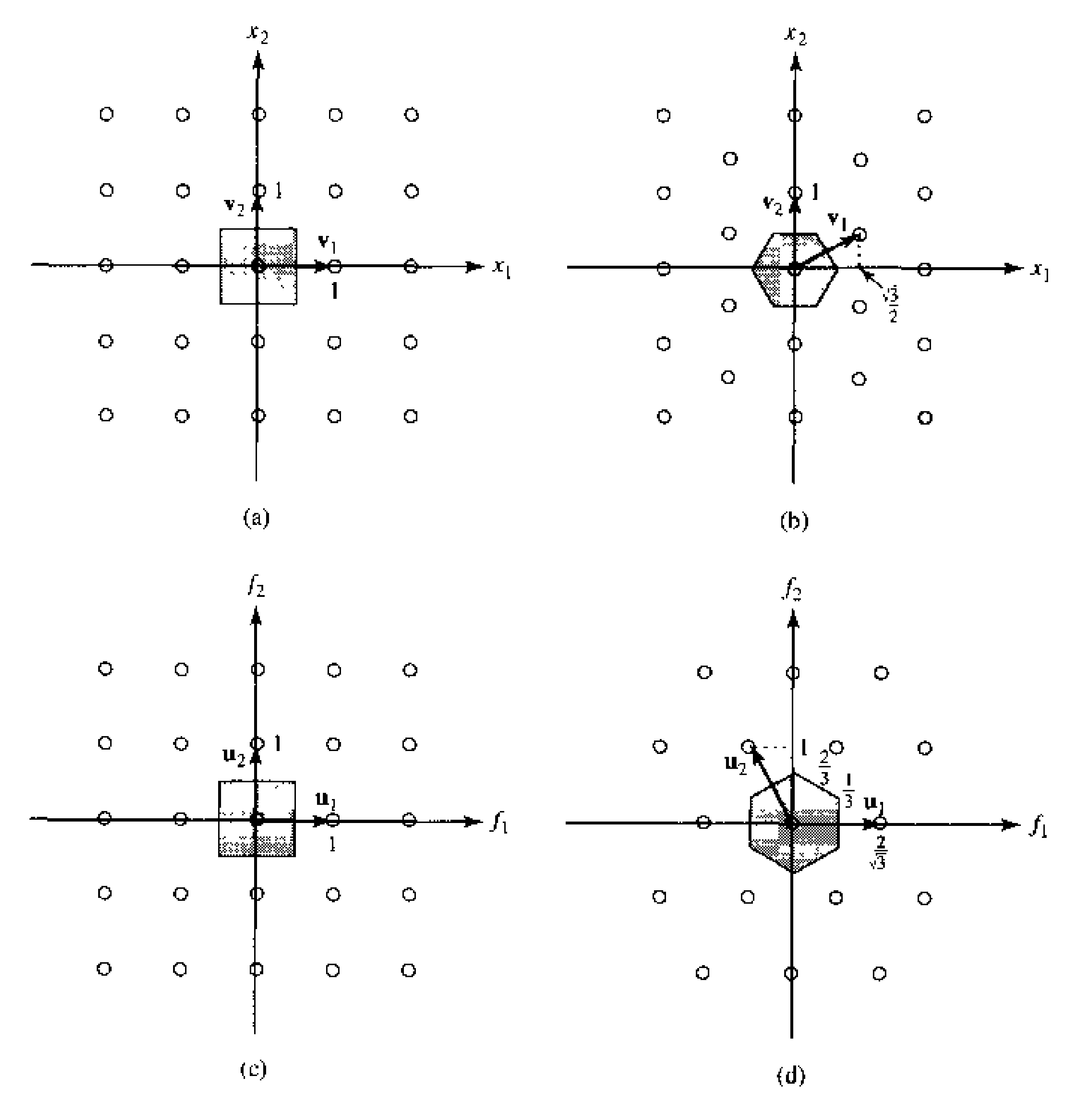


图 3.1 点阵及其反商点阵的例子：(a)矩形点阵；(b)六边形点阵；(c)矩形点阵的反商；(d)六边形点阵的反商。每个点阵的阴影部分表示该点阵的沃罗纳晶格

对于一个给定的生成矩阵，点阵中的任意一点均可由与它相关的整数系数 n_k，$k\in\mathcal{K}$标

示。使用矩阵符号,我们可以用整型矢量 $\mathbf{n}=[n_1,n_2,\cdots,n_K]^T\in\mathcal{Z}^K$ 表示点阵中任何一点。该点的实际位置是 $\mathbf{x}=[\mathbf{V}]\mathbf{n}$。点阵中任意一点均可由一个整型矢量标示的事实使点阵成为以规则几何形状采样连续信号的一个优美的工具。

下面,我们介绍点阵的一些重要性质。

定理 3.1 给定一个点阵,可以找到一个单位晶格$\mathcal{U}(\Lambda)$,它向所有阵点的平移形成对整个$\mathcal{R}^K$空间的一种分割(即非重叠覆盖);即:

$$\bigcup_{\mathbf{x}\in\Lambda}(\mathcal{U}+\mathbf{x})=\mathcal{R}^K \text{ 和 } (\mathcal{U}+\mathbf{x})\cap(\mathcal{U}+\mathbf{y})=\emptyset \quad \text{如果 } \mathbf{x}\neq\mathbf{y} \tag{3.1.3}$$

其中$\mathcal{U}+\mathbf{x}=\{\mathbf{p}+\mathbf{x}\mid\mathbf{p}\in\mathcal{U}\}$表示$\mathcal{U}$平移 $\mathbf{x}$,而 $\emptyset$ 代表空集。

定理 3.1 告诉我们$\mathcal{R}^K$空间可以由选定的单位晶格及其平移表示为一个花砖面,如图 3.2 所示。这种表示在我们考虑$\mathcal{R}^K$空间的量化时是有用的。图 3.1 所示的两种分割是均匀量化$\mathcal{R}^2$空间的两种方法。

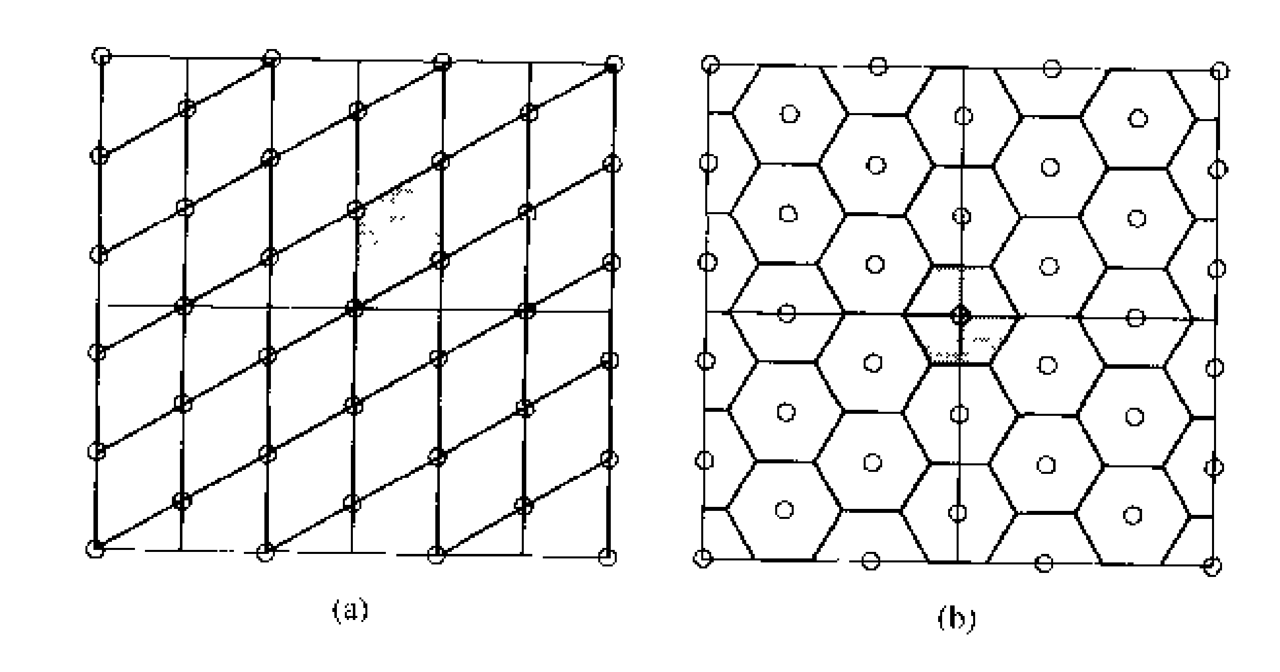


图 3.2 关于图 3.1(b)所示点阵的单位晶格:(a)基本平行四边形;(b)沃罗纳晶格。单位晶格及其向所有阵点的平移形成了对连续空间的一种分割

与一个点阵相关的单位晶格并不是惟一的。事实上,若$\mathcal{U}$是 Λ 的一个单位晶格,则$\mathcal{U}+\mathbf{x}$, $\forall\mathbf{x}\in\Lambda$ 也是一个单位晶格。在几个有趣的单位晶格中,下面定义所介绍的基本平行四边形和沃罗纳(Voronoi)晶格是最有用的。

定义 3.2 具有基矢量 $\mathbf{v}_k$,$\mathbf{k}\in\mathcal{K}$的一个点阵的基本平行四边形是由下式定义的集合:

$$\mathcal{P}(\Lambda)=\left\{\mathbf{x}\in\mathcal{R}^K\,\middle|\,\mathbf{x}=\sum_k\alpha_k\mathbf{v}_k,\forall 0\leqslant\alpha_k\leqslant 1\right\} \tag{3.1.4}$$

简言之,它是由相应于基矢量的矢量包围起来的多边形。

图 3.1(b) 给出的点阵的基本平行四边形如图 3.2(a)所示。如图中所示出的,基本平行四边形及其向所有阵点的平移形成了对空间$\mathcal{R}^K$的一种分割,因此这个基本平行四边形是一个单位晶格。显然,由于生成基的不惟一性,有很多与一个点阵相应的基本平行四边形。

定义 3.3 一个点阵的沃罗纳晶格是更接近原点而不是点阵中的任何其他点的那些点的集合。即:

$$\mathcal{V}(\Lambda)=\{\mathbf{x}\in\mathcal{R}^K\mid d(\mathbf{x},\mathbf{0})\leqslant d(\mathbf{x},\mathbf{p}),\forall\mathbf{p}\in\Lambda\} \tag{3.1.5}$$

图 3.1(b)所给出的点阵的沃罗纳晶格如图 3.2(b)所示。与基本平行四边形一样,沃罗纳晶格及其向所有阵点的平移形成了对空间$\mathcal{R}^K$的一种分割。因此沃罗纳晶格也是一个单位晶

格。我们将看到,沃罗纳晶格对于分析采样过程是非常有用的。在图3.1中,每个点阵的阴影区域指示它的沃罗纳晶格。正如所预期的,矩形点阵的沃罗纳晶格是一个简单的矩形。第二个点阵的沃罗纳晶格是一个六个边相等的六边形,这是它称为六边形点阵的原因。

在二维情况下,确定一个点阵的沃罗纳晶格,可以先在原点和每个最近的非零阵点之间画一条直线,然后在两点之间的中点画一条垂线。这条线是原点和该阵点之间的中线。所有围绕原点的这些中线所形成的多边形即为沃罗纳晶格,如图3.3所示。对于三维情况,这个过程可以通过把垂线替换为垂面进行推广。

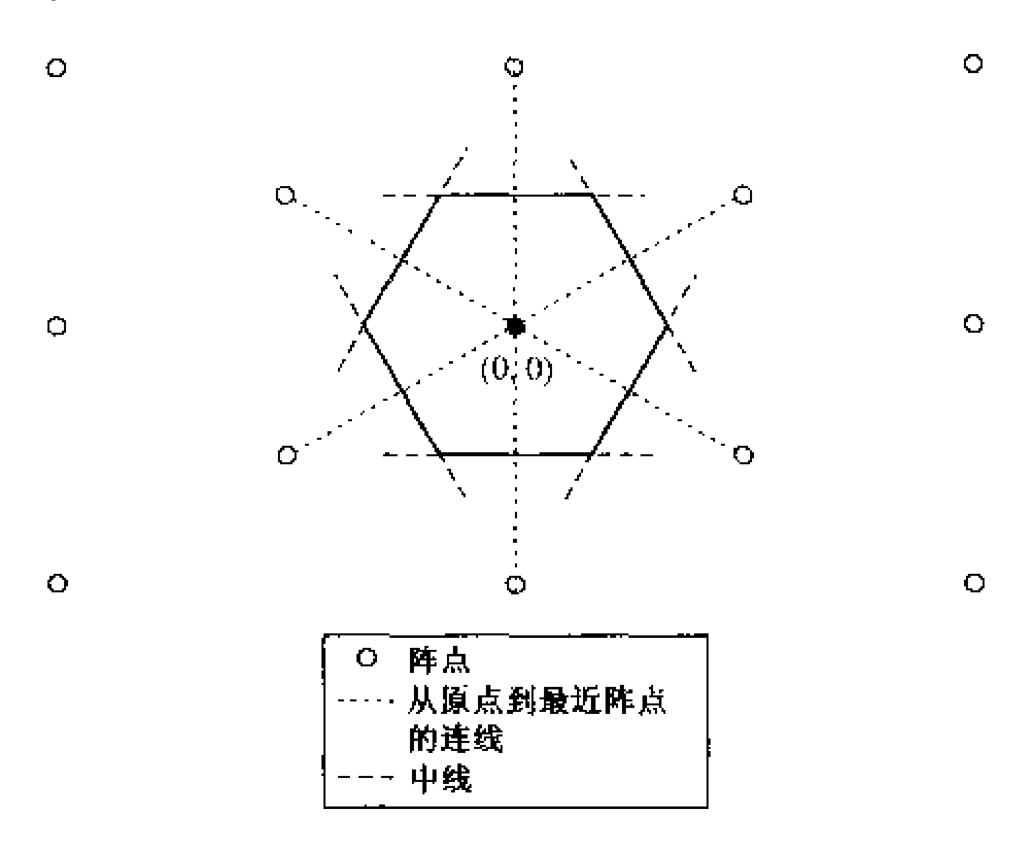


图3.3 通过画中线确定沃罗纳晶格

单位晶格的容积和采样密度 注意,尽管点阵的单位晶格并不是惟一的,单位晶格的容积却是惟一的,这是因为不管单位晶格的形状如何,需要同样数目的单位晶格去覆盖$\mathcal{R}^K$的有限子空间。根据基本代数理论,由矩阵[**V**]中的列向量形成的基本平行四边形的容积等于矩阵行列式的绝对值。因此,基本平行四边形以及其他任何单位晶格的容积均为$|\det[\mathbf{V}]|$。显然,单位晶格越小,就有越多的阵点存在于给定的容积中。因此,单位晶格容积的倒数是采样密度的度量,我们把它记为:

$$d(\Lambda) = \frac{1}{|\det[\mathbf{V}]|} \tag{3.1.6}$$

这个变量描述在$\mathcal{R}^K$的单位容积内存在多少阵点。对于图3.1(a)和(b)给出的两个点阵,容易证明它们的采样密度分别为$d_1=1$和$d_2=2/\sqrt{3}$。

定义3.4 给定一个生成矩阵为V的点阵,它的反商点阵可定义为具有如下生成矩阵的点阵:

$$[\mathbf{U}] = ([\mathbf{V}]^T)^{-1} \text{或} [\mathbf{V}]^T[\mathbf{U}] = \mathbf{I} \tag{3.1.7}$$

由定义,如果$\mathbf{x}=[\mathbf{V}]\mathbf{m}\in\Lambda$,$\mathbf{y}=[\mathbf{U}]\mathbf{n}\in\Lambda^*$,那么$\mathbf{x}^T\mathbf{y}=\mathbf{m}^T[\mathbf{V}]^T[\mathbf{U}]\mathbf{n}=\mathbf{m}^T\mathbf{n}\in\mathcal{Z}$。也就是说,分别来自两个点阵的任意两点的内积是一个整数。关系式(3.1.7)表明Λ和Λ^*的基矢量是相互正交的,即,$\mathbf{v}_k^T\mathbf{u}_l=\delta_{kl}$,$\forall k,l\in\mathcal{K}$。这里,若$k=l$,则$\delta_{kl}=1$,否则$\delta_{kl}=0$。因为$\det[\mathbf{V}]=1/\det[\mathbf{U}]$,所以两个点阵的采样密度是倒数关系,即$d(\Lambda^*)=\frac{1}{d(\Lambda)}$,这意味着$\Lambda$越密,$\Lambda^*$就

越稀疏，反之亦然。

例 3.2 对于例 3.1 给出的并示于图 3.1(a)和(b)中的两个点阵，它们的反商生成矩阵是：

$$[\mathbf{U}_1] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; [\mathbf{U}_2] = \begin{bmatrix} 2/\sqrt{3} & -1/\sqrt{3} \\ 0 & 1 \end{bmatrix} \tag{3.1.8}$$

基于$[\mathbf{U}_i]$中的基矢量，我们可以用与确定原始点阵中的点同样的方法来确定反商点阵中所有的点。结果如图 3.1(c)和(d)所示。反商点阵的沃罗纳晶格也在这两个图中给出。我们看到，对于矩形点阵，原始点阵和反商点阵的沃罗纳晶格均为矩形，而对于六边形点阵，原始和反商沃罗纳晶格均为六边形，但具有不同的取向。我们还可以看出，Λ_2 比 Λ_1 密集，而 Λ_2^* 比 Λ_1^* 稀疏。

定理 3.2 令 Λ_1 和 Λ_2 分别代表具有生成矩阵$[\mathbf{V}_1]$和$[\mathbf{V}_2]$的两个点阵，如果$[\mathbf{V}_1]^{-1}[\mathbf{V}_2]$是一个有理数矩阵，那么交集：

$$\Lambda_1 \cap \Lambda_2 = \{\mathbf{x} \in \mathcal{R}^K \mid \mathbf{x} \in \Lambda_1 \text{ 且 } \mathbf{x} \in \Lambda_2\}$$

也是一个点阵，并且它是包含于 Λ_1 和 Λ_2 二者中的最大的点阵。在同样条件下，Λ_1 与 Λ_2 的和：

$$\Lambda_1 + \Lambda_2 = \{\mathbf{x} + \mathbf{y} \mid \forall \mathbf{x} \in \Lambda_1, \forall \mathbf{y} \in \Lambda_2\}$$

也是一个点阵，它是包含于 Λ_1 和 Λ_2 二者中的最小的点阵。进一步，$(\Lambda_1 + \Lambda_2)^* = \Lambda_1^* \cap \Lambda_2^*$。

定理 3.3 若 $\Lambda_1 \subset \Lambda_2$，则 $\Lambda_1^* \supset \Lambda_2^*$。或者写成，若 $\Lambda_1 \subset \Lambda_2$，则$\mathcal{V}(\Lambda_1) \supset \mathcal{V}(\Lambda_2)$。

上面的结果可以通过这样的事实证明：点阵必须包含能表示为$[\mathbf{V}]\mathbf{n}$的点，其中 $\mathbf{n} \in \mathcal{Z}^K$，而$[\mathbf{V}]$是一个非奇异矩阵。在 4.1 节讨论采样率转换时，这些定理的用处将会显现出来。

用点阵概念表征周期性 我们都很熟悉一维周期性的概念。如果$\psi(x) = \psi(x + nT)$，$\forall n \in \mathcal{Z}$，我们说一个函数是以 T 为周期的周期函数。点阵概念的一个重要应用是描述多维函数的周期性，定义如下。

定义 3.5 如果对于任意 $\mathbf{n} \in \mathcal{Z}^K$ 有$\psi(\mathbf{x}) = \psi(\mathbf{x} + [\mathbf{V}]\mathbf{n})$，则一个函数是周期性的，且具有非奇异的周期矩阵$[\mathbf{V}]$。

显然，所有重复中心的集合$[\mathbf{V}]\mathbf{n}, \mathbf{n} \in \mathcal{Z}^K$，形成了一个具有生成矩阵$[\mathbf{V}]$的点阵。因此，当用一个点阵来描述一个函数的周期性时，它也称为周期性点阵。我们记得，整个$\mathcal{R}^K$ 可以被分割成无穷多个该点阵单位晶格的移位副本。因此，一个周期函数可以认为是在所有其他移位的单位晶格处重复所选择的单位晶格的基本图形。我们称该点阵的沃罗纳晶格为这个函数的基本周期。对于周期为 T 的一维函数，基本周期是中心位于原点的时间间隔$(-T/2, T/2)$。一个周期函数实质上是由基本周期以及这个周期在所有阵点上的移位副本组成的。显然只需要在它的基本周期上规定一个周期函数。

3.2 在点阵上采样

点阵结构为用一个均匀的但不一定是超立方体的格栅采样连续信号提供了一个漂亮的工具。它使我们能够用整数矢量为所有的样点标号，简化了对采样后所产生的信号的描述。在点阵上采样是在二维的矩形格栅或 K 维的超立方体格栅上均匀采样的一个推广。正如将要

说明的，使用这种采样结构，所有已知的一维和二维均匀采样定理仍然成立。尤其是，存在推广的奈奎斯特采样定理，它决定了对于给定的信号频谱，采样点阵所需要的密度和结构。

3.2.1 采样过程和采样空间的傅里叶变换

定义 3.6 给定一个连续信号$\psi_c(\mathbf{x}),\mathbf{x}\in\mathcal{R}^K$，在生成矩阵为$[\mathbf{V}]$的点阵$\Lambda$上的采样信号定义为：

$$\psi_s(\mathbf{n}) = \psi_c([\mathbf{V}]\mathbf{n}),\mathbf{n}\in\mathcal{Z}^K \tag{3.2.1}$$

另一种定义采样信号的方法是把它当做在采样点具有冲激的连续空间信号，即：

$$\psi_s(\mathbf{x}) = \sum_{\mathbf{n}\in\mathcal{Z}^K}\psi_c(\mathbf{x})\delta(\mathbf{x}-[\mathbf{V}]\mathbf{n}),\mathbf{x}\in\mathcal{R}^K \tag{3.2.2}$$

在2.2节里我们定义了离散空间傅里叶变换。类似地，我们可以定义在一个点阵上采样信号的傅里叶变换[5,4]。这里我们称之为采样空间傅里叶变换(SSFT)。与通过采样点抽象的标号定义的DSFT相比，SSFT的定义保持了图像中离散信号的潜在采样结构，并且使我们能够与实际的物理维建立联系。

定义 3.7 对于在一个具有生成矩阵$[\mathbf{V}]$的点阵Λ上的被采样信号，SSFT定义为：

$$\Psi_s(\mathbf{f}) = \sum_{\mathbf{n}\in\mathcal{Z}^K}\psi_s(\mathbf{n})\exp(-\mathrm{j}2\pi\mathbf{f}^T[\mathbf{V}]\mathbf{n}) \tag{3.2.3}$$

利用公式(3.2.2)中的定义，可以证明采样信号的CSFT等于它的SSFT。因此，SSFT定义与CSFT定义一致。而且，当点阵是超立方体时，SSFT简化为DSFT，此时$[\mathbf{V}]$是一个K维恒等矩阵。

注意到若$\mathbf{f}^T\mathbf{x}=n\in\mathcal{Z}$，$\exp(\mathrm{j}2\pi\mathbf{f}^T\mathbf{x})=1$，我们有：

$$\Psi_s(\mathbf{f}+[\mathbf{U}]\mathbf{m}) = \Psi_s(\mathbf{f}),[\mathbf{U}] = ([\mathbf{V}]^T)^{-1}$$

这意味SSFT是周期性的，以周期性矩阵$[\mathbf{U}]$为周期，并且频谱重复中心落在采样点阵的反商Λ^*上。基本周期是反商点阵的沃罗纳晶格$\mathcal{V}(\Lambda^*)$。因此，只需确定在$\mathcal{V}(\Lambda^*)$上的被采样信号的SSFT，这个基本图形在反商点阵的所有点上重复(在3.2.2节对此做了详细的描述和图示)。

定理 3.4(逆采样空间傅里叶变换) 一个在生成矩阵为$[\mathbf{V}]$的点阵Λ上的被采样信号，可以由它的SSFT得到，即：

$$\psi_s(\mathbf{n}) = \frac{1}{d(\Lambda)}\int_{\mathcal{V}(\Lambda^*)}\Psi_s(\mathbf{f})\exp(\mathrm{j}2\pi\mathbf{f}^T[\mathbf{V}]\mathbf{n})d\mathbf{f},\ \mathbf{n}\in\mathcal{Z}^K \tag{3.2.4}$$

证明这个反变换公式的正确性，可以通过用公式(3.2.3)中的正变换代替公式(3.2.4)中的$\Psi_s(\mathbf{f})$，并使用等式：

$$\int_{\mathcal{V}(\Lambda^*)}\exp(\mathrm{j}2\pi\mathbf{x}^T\mathbf{f})d\mathbf{f} = \begin{cases}\text{volume}\{\mathcal{V}(\Lambda^*)\} = d(\Lambda), & \mathbf{x}=\mathbf{0}\\ 0, & \text{其他}\end{cases} \tag{3.2.5}$$

公式(3.2.4)中的反变换表明，一个采样信号可以表示为复指数函数的无限求和，复指数函数的频率以反商点阵的沃罗纳晶格定义。

CSFT的所有性质均可转移给SSFT。我们这里只给出卷积定理。定理的证明是简单的，并留做一个练习(习题3.4)。

定义 3.8 在同一个点阵上采样的两个信号的卷积定义为：

$$\psi_s(\mathbf{n})*h_s(\mathbf{n}) = \sum_{\mathbf{m}\in\mathcal{Z}^K}\psi_s(\mathbf{n}-\mathbf{m})h_s(\mathbf{m}) = \sum_{\mathbf{x}_m\in\Lambda}\psi_c([\mathbf{V}](n-m))h_c([\mathbf{V}]m) \tag{3.2.6}$$

其中$\psi_c(\mathbf{x})$和$h_c(\mathbf{x})$分别是$\psi_s(\mathbf{n})$和$h_s(\mathbf{n})$的连续函数。

定理 3.5(采样空间卷积定理) 与 CSFT 和 DSFT 一样，采样空间中两个信号的卷积等价于它们的 SSFT 的乘积，即：

$$\psi_s(\mathbf{n}) * h_s(\mathbf{n}) \leftrightarrow \Psi_s(\mathbf{f}) H_s(\mathbf{f}) \tag{3.2.7}$$

采样线性移不变系统的表征 给定一个在连续空间$\mathcal{R}^K$上的冲激响应为$h_c(\mathbf{x})$的线性移不变系统，若对输入和输出信号在点阵Λ上采样，则可以证明(习题 3.5)，被采样的输入信号$\psi_s(\mathbf{n}) = \psi_c([\mathbf{V}]\mathbf{n})$与输出信号$\phi_s(\mathbf{n}) = \phi_c([\mathbf{V}]\mathbf{n})$，具有前面定义的在$\Lambda$上的卷积关系，而且滤波器恰恰是被采样的脉冲响应$h_s(\mathbf{n}) = h_c([\mathbf{V}]\mathbf{n})$。因此，被采样的系统完全由被采样的脉冲响应$h_s(\mathbf{n})$表征。

3.2.2 推广的奈奎斯特采样定理

定理 3.6(推广的奈奎斯特采样定理) 如果一个连续信号$\psi_c(\mathbf{x}), \mathbf{x} \in \mathcal{R}^K$在生成矩阵为$[\mathbf{V}]$的点阵$\Lambda$上被采样，那么采样信号的 SSFT，$\psi_s(\mathbf{n}) = \psi_c([\mathbf{V}]\mathbf{n}), \mathbf{n} \in \mathcal{Z}^K$是原始连续信号的 CSFT 及其在反商点阵$\Lambda^*$的所有点上的移位副本的总和，再乘以点阵的密度，即：

$$\Psi_s(\mathbf{f}) = d(\Lambda) \sum_{\mathbf{m} \in \mathcal{Z}^K} \Psi_c(\mathbf{f} - [\mathbf{U}]\mathbf{m}) \tag{3.2.8}$$

当且只当原始信号的 CSFT 的非零区(称为支撑区)在反商点阵的沃罗纳晶格内是有限的，才能够从被采样信号完全恢复原始连续信号，即：

$$\Psi_c(\mathbf{f}) = 0, \ \mathbf{f} \notin \mathcal{V}(\Lambda^*) \tag{3.2.9}$$

进而，可以通过用具有如下频率响应(按 CSFT 定义)的重建滤波器对采样信号进行滤波实现完全重建：

$$H_r(\mathbf{f}) = \begin{cases} \dfrac{1}{d(\Lambda)}, & \mathbf{f} \in \mathcal{V}(\Lambda^*) \\ 0, & \text{其他} \end{cases} \tag{3.2.10}$$

容易看出，定理 3.6 是一维奈奎斯特采样定理的推广。在那个特殊情况下，$[\mathbf{V}] = [\Delta]$，$[\mathbf{U}] = [1/\Delta]$，并且$d(\Lambda) = 1/\Delta$，这里$\Delta$为采样间隔。与一维和二维信号一样，我们把重复频谱称为混叠分量。这个定理告诉我们，如果Λ是空间域采样点阵，那么在频域Λ^*将环绕混叠分量的各个中心。如果连续信号的支撑区大于$\mathcal{V}(\Lambda^*)$，那么相邻的混叠分量将会重叠使原始高频分量折叠到低频区。这个效应称为混叠。我们已经知道，Λ越密集，Λ^*就越稀疏。因此，使用较密集的采样格栅会进一步分离频域内的混叠分量，从而减少被采样信号的混叠效应。

对于一个固定的采样点阵，为了避免混叠，原始的连续信号应该用如下预滤波器进行带限：

$$H_p(\mathbf{f}) = \begin{cases} 1, \mathbf{f} \in \mathcal{V}(\Lambda^*) \\ 0, \text{其他} \end{cases} \tag{3.2.11}$$

我们看到，预滤波器和重建滤波器在理想情况下都应是具有由反商点阵的沃罗纳晶格$\mathcal{V}(\Lambda^*)$定义支撑区的低通滤波器。

为证明定理 3.6，我们可以利用逆 CSFT。由公式 (2.1.7)有：

$$\psi_s(\mathbf{n}) = \psi_c([\mathbf{V}]\mathbf{n}) = \int_{\mathcal{R}^K} \Psi_c(\mathbf{f}) \exp(j2\pi \mathbf{f}^T [\mathbf{V}]\mathbf{n}) d\mathbf{f}$$

$$= \sum_{\mathbf{m}\in \mathcal{Z}^K}\int_{\mathcal{V}(\Lambda^*)} \Psi_c(\mathbf{f}-[\mathbf{U}]\mathbf{m})\exp(j2\pi(\mathbf{f}-[\mathbf{U}]\mathbf{m})^T[\mathbf{V}]\mathbf{n})\,d\mathbf{f}$$

$$= \sum_{\mathbf{m}\in \mathcal{Z}^K}\int_{\mathcal{V}(\Lambda^*)} \Psi_c(\mathbf{f}-[\mathbf{U}]\mathbf{m})\exp(j2\pi\mathbf{f}^T[\mathbf{V}]\mathbf{n})\,d\mathbf{f}$$

$$= \int_{\mathcal{V}(\Lambda^*)}\sum_{\mathbf{m}\in \mathcal{Z}^K} \Psi_c(\mathbf{f}-[\mathbf{U}]\mathbf{m})\exp(j2\pi\mathbf{f}^T[\mathbf{V}]\mathbf{n})\,d\mathbf{f}$$

把最后一个等式与公式(3.2.4)比较,就会得到公式(3.2.8)的结果。

例3.3 考虑一个具有图3.4(a)所示频谱的二维信号,它具有半径 $r=1/\sqrt{3}$的圆形支撑区。如果我们用图3.1(a)所给出的点阵采样该信号,采样信号的频谱将如图3.4(b)所示,它是通过在图3.1(c)所示的反商点阵的各点重复原始频谱得到的。显然,由于信号的支撑区大于反商点阵的沃罗纳晶格,在采样信号频谱中的混叠分量有重叠。为避免重叠,我们可以对连续信号进行预滤波,使其频谱只在沃罗纳晶格内是非零的,如图3.4(c)所示。如果我们现在用同样的点阵对这个信号进行采样,所有混叠分量将无重叠地精确覆盖整个频谱,如图3.4(d)所示。用理想重建滤波器重建的信号将精确地与预滤波后的信号相同,如图3.4(c)所示。显然,这个信号与原始信号相比,丢失了某些频谱成分。如果在应用中要求保留原始信号的全部带宽,那么就需要采样点阵具有一个等于或大于信号支撑区的反商沃罗纳晶格。例如,如果我们使用图3.1(b)所示的六边形点阵,那么信号频谱正好与它的反商点阵的沃罗纳晶格相配合,如图3.4(e) 所示,采样信号的频谱将如图3.4(f)所示不发生混叠。在这种情况下,重建信号将是完全的。不用图3.1(b)所给出的六边形点阵,我们可以仍然使用一个具有较小采样间隔 $\Delta_1=\Delta_2=\sqrt{3}/2<1$ 的矩形点阵。这将导致一个较高的采样密度。回想我们前面导出第一个和第二个点阵的采样密度分别是1和 $2/\sqrt{3}$,而前面那个缩小了的矩形点阵的采样密度为4/3,高于六边形点阵。从这个例子可以看出,六边形点阵比矩形点阵更有效,因为要得到无混叠采样,它只需要较低的采样密度。

3.2.3 采样效率

如例3.3所说明的,为了避免混叠,设计采样点阵时必须使其反商点阵的沃罗纳晶格完全覆盖信号的频谱。显然,总可以设计一个非常密集的点阵来满足这一要求,然而,这将要求很高的采样密度。为了最小化采样密度,点阵的设计应使其反商沃罗纳晶格尽可能紧凑地覆盖信号频谱。解决方案将依赖于信号的频谱,它一般可能具有一个任意形状的支撑区。

幸运的是,大部分现实世界的信号在不同方向上(适当标度频率轴以后)的频谱成分是对称的,它们的频谱支撑区可以被很好地近似为球形。因此,为了比较不同点阵的采样效率,我们可以计算一个具有半径为1的球形支撑区的信号所需的采样密度。为了避免混叠,采样点阵反商的沃罗纳晶格$\mathcal{V}(\Lambda^*)$必须包围这个球。$\mathcal{V}(\Lambda^*)$越紧凑地充满球,所需要的采样密度就越低。这导致采样效率[①] 定义如下:

① 采样效率的另外一种定义是将$\mathcal{V}(\Lambda^*)$的体积与覆盖球的单位立方体的体积比较得到的,即:

$$\rho(\Lambda)=\frac{\text{volume}(\mathcal{V}(\Lambda^*))}{\text{volume}(\text{单位立方体})}$$

根据这个定义,较小的值表示较高的有效性。

$$\rho(\Lambda)=\frac{\text{volume}(\text{单位球})}{\text{volume}(\mathcal{V}(\Lambda^*))}=\frac{\text{volume}(\text{单位球})}{d(\Lambda)} \tag{3.2.12}$$

上面的值越接近1,点阵就越有效。

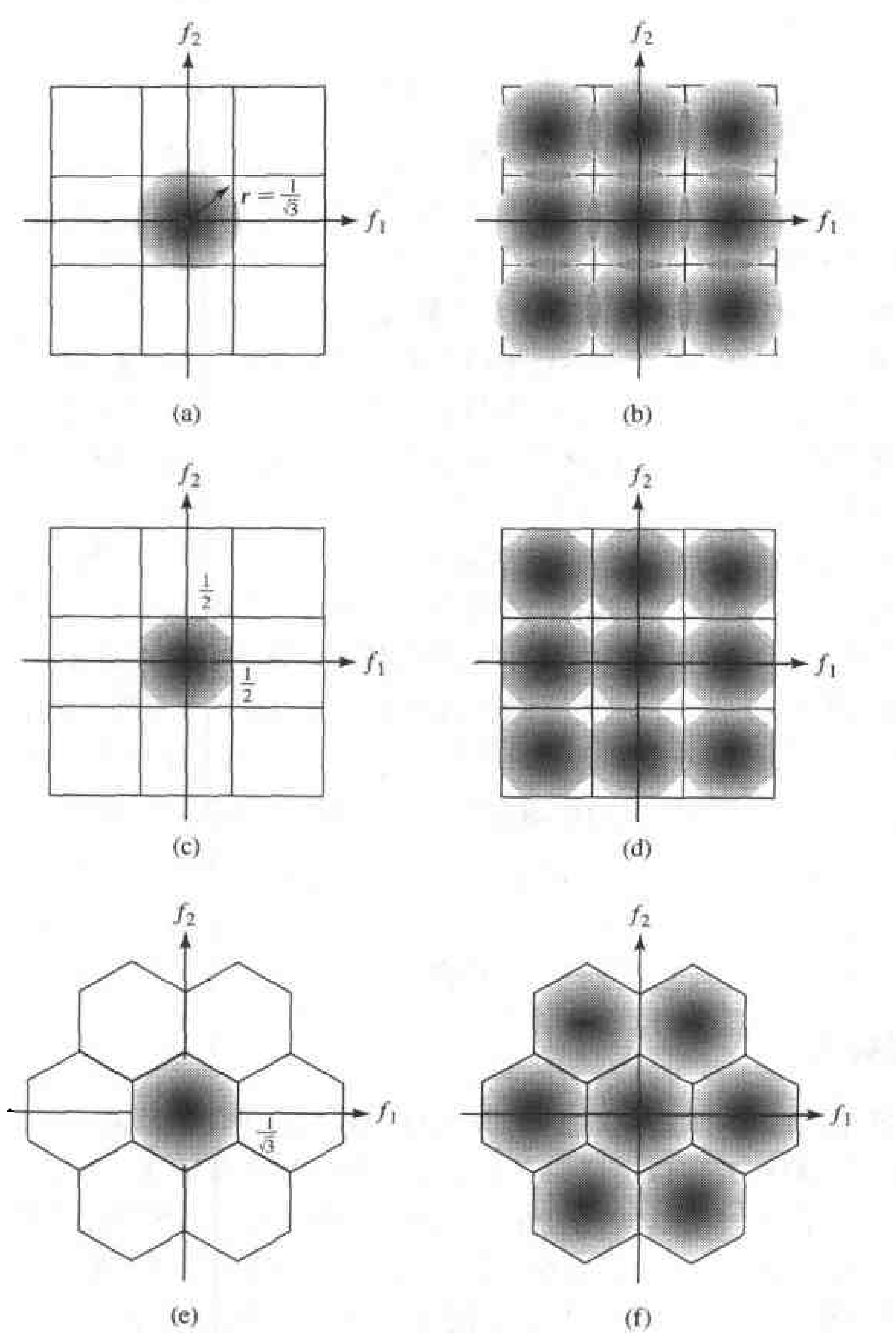


图3.4 频域采样过程图示:(a)原始信号;(b)用图3.1(c)所示的矩形点阵采样的信号;(c)预滤波的信号;(d)经过预滤波的采样信号;(e)铺在图3.1(d)所示的六边形分割上的原始信号的频谱;(f)使用六边形点阵的采样信号。本图中最深的阴影代表频谱的最高幅度

我们记得,单位球的体积在二维情况下为π,三维情况下为4/3π。另外一方面,覆盖球的最小立方体的体积在二维情况下为4,三维情况下为8。因此,立方体点阵的效率在二维情况下为π/4,三维情况下为π/6。因此立方点阵对于采样具有球形支撑的信号不是很有效。

图3.5示出了覆盖球形的几种二维点阵的效率。可以看出,六边形点阵覆盖球最紧凑。因此,采样效率也可以相对于六边形点阵的密度进行定义,这时六边形点阵的效率变为1。

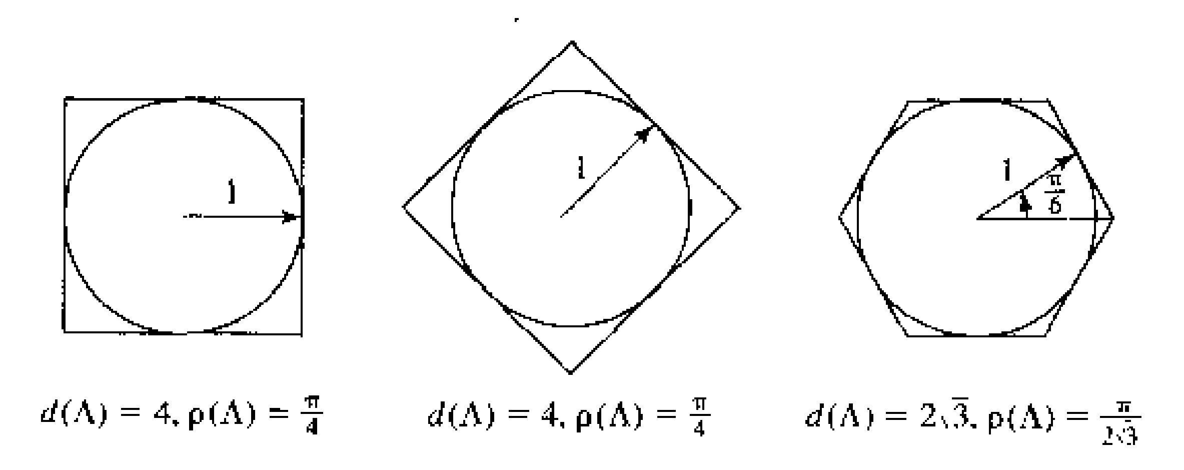


图 3.5 二维矩形、菱形和六边形点阵的球形覆盖能力的图示

3.2.4 预滤波器和重建滤波器的实现

在前面的讨论中,我们已经在频域描述了理想的预滤波器和重建滤波器。采用逆 CSFT,可以得到空间域的等效滤波器。尽管预滤波操作必须完全在连续域进行,但重建操作可以简化,因为输入信号只在被采样的点上存在。为了导出空间域的等价内插运算,让我们把采样后的信号认为是在采样点上有冲激的连续信号,正如公式(3.2.2)所描述的。于是滤波运算可以简化为:

$$\begin{aligned}\psi_r(\mathbf{x}) &= \int_{\mathbf{y}\in\mathcal{R}^K} h_r(\mathbf{x}-\mathbf{y})\psi_s(\mathbf{y})d\mathbf{y} \\ &= \sum_{\mathbf{n}\in\mathcal{Z}^K}\psi_s(\mathbf{n})\int_{\mathbf{y}\in\mathcal{R}^K} h_r(\mathbf{x}-\mathbf{y})\delta(\mathbf{y}-[\mathbf{V}]\mathbf{n})d\mathbf{y} \\ &= \sum_{\mathbf{n}\in\mathcal{Z}^K}\psi_s(\mathbf{n})h_r(\mathbf{x}-[\mathbf{V}]\mathbf{n})\end{aligned} \tag{3.2.13}$$

这样在 $\mathbf{x}$ 点的内插值是所有采样值的加权和,对于样点 $\mathbf{n}$ 的权值为 $h_r(\mathbf{x}-[\mathbf{V}]\mathbf{n})$。这是众所周知的一维内插公式的 K 维形式。

注意,由于期望频率响应中的锐过渡带,理想的预滤波器和内插滤波器将具有无限的空间扩展。实际上,只有有限扩展的滤波器是可实现的。我们将在 4.1.4 节中讨论滤波器设计的问题。

3.2.5 连续、离散和采样空间傅里叶变换之间的关系

注意到被采样信号 $\psi_s(\mathbf{n})$ 是 2.2 节所述的离散空间信号的一种特殊情况。因此,可以把公式(2.2.3)所定义的 DSFT 应用于在任意点阵上被采样的信号。与 SSFT 比较,DSFT 没有考虑到实际采样的几何结构。由定义,我们有:

$$\Psi_s(\mathbf{f}) = \sum_{\mathbf{n}\in\mathcal{Z}^K}\psi_s(\mathbf{n})\exp(-j2\pi\mathbf{f}^T[\mathbf{V}]\mathbf{n}) \tag{3.2.14}$$

$$\Psi_d(\mathbf{f}) = \sum_{\mathbf{n}\in\mathcal{Z}^K}\psi_s(\mathbf{n})\exp(-j2\pi\mathbf{f}^T\mathbf{n}) \tag{3.2.15}$$

比较这两个公式,我们看到:

$$\Psi_d(\mathbf{f}) = \Psi_s([\mathbf{U}]\mathbf{f}), \mathbf{f}\in\mathcal{I}^K, \text{和 } \Psi_s(\mathbf{f}) = \Psi_d([\mathbf{V}]^T\mathbf{f}), \mathbf{f}\in\mathcal{V}(\Lambda^*) \tag{3.2.16}$$

这告诉我们,在单位超立方体 $\mathcal{I}^K$ 中的DSFT是在 $\mathcal{I}^K$ 的扭曲 $\mathcal{V}(\Lambda^*)$ 上的SSFT图像,或在 $\mathcal{V}(\Lambda^*)$ 中的

SSFT是在$\mathcal{V}(\Lambda^*)$的扭曲$\mathcal{I}^K$上的DSFT的图像①。离散频率变量$\mathbf{f}_d$与连续频率变量的关系为：

$$\mathbf{f}_c = [\mathbf{U}]\mathbf{f}_d = ([\mathbf{V}]^T)^{-1}\mathbf{f}_d, \text{和 } \mathbf{f}_d = [\mathbf{V}]^T\mathbf{f}_c = [\mathbf{U}]^{-1}\mathbf{f}_c \tag{3.2.17}$$

回忆公式(3.2.8)中$\Psi_s(\mathbf{f})$与$\Psi_c(\mathbf{f})$之间的关系，我们有：

$$\Psi_d(\mathbf{f}) = d(\Lambda)\sum_{\mathbf{m}\in\mathcal{Z}^K}\Psi_c([\mathbf{U}]\mathbf{f} - [\mathbf{U}]\mathbf{m}) \tag{3.2.18}$$

如果$\psi_c(\mathbf{x})$在$\mathcal{V}(\Lambda^*)$内是带限的，那么：

$$\Psi_d(\mathbf{f}) = d(\Lambda)\Psi_c([\mathbf{U}]\mathbf{f}), \mathbf{f}\in\mathcal{I}^K \tag{3.2.19}$$

显然，$\Psi_d(\mathbf{f})$是周期性的，其重复中心位于所有的整数点上，而这些点对应于连续频域中Λ^*内的点。

我们还记得，在一维情况下，如果我们以采样频率f_s采样连续信号，离散频率和连续频率的关系是$f_d = f_c/f_s$，并且离散频域的基本周期$(-1/2, 1/2)$对应连续频域的基本周期$(-f_s/2, f_s/2)$。这是此处对于K维信号所描述的情况的一个特例，其中$[\mathbf{V}] = [1/f_s]$，$[\mathbf{U}] = [f_s]$。

3.3 视频信号的采样

前面各节给出了一般K维信号的采样定理。在本章剩下的各节中，我们重点讨论三维信号的采样。两个空间维和时间维是不对称的，因为它们有不同的特性，并且对于空间和时间频率的视觉敏感度是不同的。这种不对称性导致了几种有趣的视频采样技术的提出。

尽管视频信号在空间和时间上是连续变化的，但今天的摄像机还不能在所有三维上完全连续地摄取信号。大多数运动图片摄像机在时间方向上对景物进行采样，并把一个模拟图像帧的序列存储在胶片上。另一方面，大多数TV摄像机是通过在时间和垂直方向进行采样来摄取视频序列的。产生的信号存储在一个一维光栅扫描中，它是沿着相继的水平扫描行的连成一串的颜色变化。为了获得一个完全的数字视频，可以对运动图片摄像机产生的模拟帧进行二维采样，或对TV摄像机产生的模拟光栅扫描进行一维采样，或用数字摄像机对景物进行三维采样直接获得离散视频帧。不同的采样方案对应不同的采样点阵。在设计数字视频采样系统时，必须回答的两个问题是：(1)所需的视频采样率是多少？(2)在给定总的采样率时，哪一种采样点阵最有效？本节先描述确定所需的视频采样率的因素，然后讨论二维采样方案，并且对逐行扫描和隔行扫描进行比较。接下来，介绍几个三维采样点阵并对它们的效率进行比较。最后，举一个特殊的例子说明空间和时间混叠效应。

3.3.1 所需要的采样率

在设计视频采样系统时，无论一维、二维还是三维，一个必须回答的基本问题是，空间和时间采样的分辨率应该是多少？答案由几个因素决定：待采样信号的频率成分，在空间和时间截止频率方面的视觉阈值，摄取和显示设备的特性，以及可承受的处理、存储和传输成本。基于采样定理，如果我们采用立方体点阵，每维上的采样率至少应为该方向上最高频率的两倍。另一方面，从2.4节可知，人眼不能分辨超过一定高的频率的空间和时间变化。尽管信号的最高

① 注意，相应于$[\mathbf{U}]\mathbf{f}, \mathbf{f}\in\mathcal{I}^K$的空间区域，一般并不对应$\mathcal{V}(\Lambda^*)$。但是由于$\Psi_c(\mathbf{f})$和$\Psi_d(\mathbf{f})$都是周期性的，函数$\Psi_d(\mathbf{f}), \mathbf{f}\in\mathcal{I}^K$来自于$\mathcal{V}(\Lambda^*)$的不同部分。在习题3.6中对此做了进一步的考虑。

频率可能变化相当大，而视觉截止频率，即可以被 HVS 观察到的空间和时间的最高频率，应该是确定视频采样率的决定因素，不需容纳这些值以外的频率成分。有理由假设，技术总是能被推进以适应人类的需求。

如 2.4 节所述，视觉敏感度依赖于显示器的平均亮度。对于很亮的 TV 信号，视觉阈值要求帧率大于 70 Hz，而空间分辨率至少是 30 cpd。在屏幕高度三倍的正常观测距离上，25 cpd 的空间频率换算为大约 500 行/帧(lpf)。对于每一行采样，水平采样间隔应大致等于垂直间隔，这样产生的像素是方形的(即 PAR 为 1)。这导致大约 670 像素/行，共显示 500 行，IAR 为 4:3。

在模拟 TV 系统被开发出来的时候，视觉截止频率所要求的这些采样率超过了当时的技术水平。为了降低码率进而降低视频摄取、传输和显示的成本，隔行扫描被提出来了，对于给定的总码率(帧率和行率的乘积)，它靠牺牲垂直分辨率来提高时间分辨率。例如，如 1.4 节所讲的，在 NTSC TV 系统中，每秒摄取 60 个场，但每场只包含所期望行数的一半(240 有效行/场)。与采用 30 fps 和 480 lpf 的逐行扫描的总码率是相同的。如果景物是静止的，通过每一场扫描线的交错，它能够产生与采用 60 fps 和 480 lpf 逐行扫描一样的质量。但对于具有特殊图案(垂直线图案)的高速运动场景，它会导致所谓的“隔行效应”。下一代数字 TV 系统运行在 30 fps 和 480 lpf 方式或 60 场/秒和 240 行/场方式。隔行格式的保留主要是为了与模拟 TV 系统兼容。HDTV 系统进一步加强视觉效果，采用 IAR 为 16:9，采样分辨率为 60 fps 和 720 lpf 的较宽的屏幕。同样为了兼容目的，也可以采用 60 场/秒和 540 行/场的隔行格式(见表 1.3)。

对于运动图片，由于电影院的环境亮度很低，降低了视觉敏感性，24 fps 的帧率(逐行)就足够了。如 2.4.1 节所述，尽管原始图像是在 24 fps 帧率下摄取的，但在回放时，在投射镜头前放置一个每帧旋转 3 次的叶片，使得有效回放速率为 72 fps。这就抑制了可能会被一些敏感的观众感受到的闪烁效应。许多老的运动图片是用 IAR 为 4:3 的屏幕放映的，为了更加戏剧性的视觉效果，较新的运动图片通常是采用 IAR 高达 2:1 放映的。

对于计算机显示，需要高得多的空间和时间采样率。例如，SVGA 显示的帧率为 72 fps(逐行)，空间分辨率为 1 024 × 720 像素。这是为了适应很近的观察距离(通常为图片高度的一到二倍)和显示资料的高频成分(线图形和文本)。

3.3.2 二维视频采样:逐行和隔行扫描

在 1.3 节描述了光栅扫描机制，包括逐行和隔行扫描。需要搞清楚的是视频光栅实际上是在时间和垂直方向上被采样的一种三维视频信号的形式。正如已经提到的，采用隔行扫描的动机是，当在给定时间内能够记录的总行数给定时，用牺牲垂直分辨率来换取增强的时间分辨率。在本节中，我们通过分析这两种采样策略的频域混叠图形来更深入地了解它们相应的显示效应。

为了便于讨论，我们忽略水平方向，并且把视频信号看做是在由时间和垂直方向生成的空间中的二维信号。令 Δ_t 表示场间隔，而 Δ_y 表示行间隔。这样，逐行和隔行扫描所使用的采样点阵分别如图 3.6(a) 和 (b) 所示。图中也标示了生成每个点阵的基矢量。由这些基矢量，可以导出原始点阵和反商点阵的如下生成矩阵：

$$\text{逐行扫描}:[\mathbf{V}_1]=\begin{bmatrix}2\Delta_t & 0\\ 0 & \Delta_y\end{bmatrix},[\mathbf{U}_1]=\begin{bmatrix}1/2\Delta_t & 0\\ 0 & 1/\Delta_y\end{bmatrix} \tag{3.3.1}$$

$$\text{隔行扫描}:[\mathbf{V}_2]=\begin{bmatrix}2\Delta_t & \Delta_t\\ 0 & \Delta_y\end{bmatrix},[\mathbf{U}_2]=\begin{bmatrix}1/2\Delta_t & 0\\ -1/2\Delta_y & 1/\Delta_y\end{bmatrix} \tag{3.3.2}$$

基于矩阵$[\mathbf{U}_1]$和$[\mathbf{U}_2]$，可以画出图3.6(c)和(d)中的反商点阵。其中我们加深了频率平面第一象限内离原点最近的3个点(实心圆圈)。这几点是最接近原始信号频谱的混叠分量的中心，也是可觉察失真的主要原因。

注意，画点阵时，我们改变了空间和时间维的比例，使得等于垂直采样率的空间频率$f_{s,y}=1/\Delta_y$与等于场率的时间频率$f_{s,t}=1/\Delta_t$具有相同的长度。类似地，空间间隔Δ_y和时间间隔Δ_t也具有相同的长度。理想情况下，通过给空间和时间截止频率分配相同的长度，我们愿意将基于视觉敏感度的时间和空间频率等同对待。因为空间和时间采样频率通常是基于各自的截止频率选取的，使不同方向上的采样频率相等一般来说是合适的。

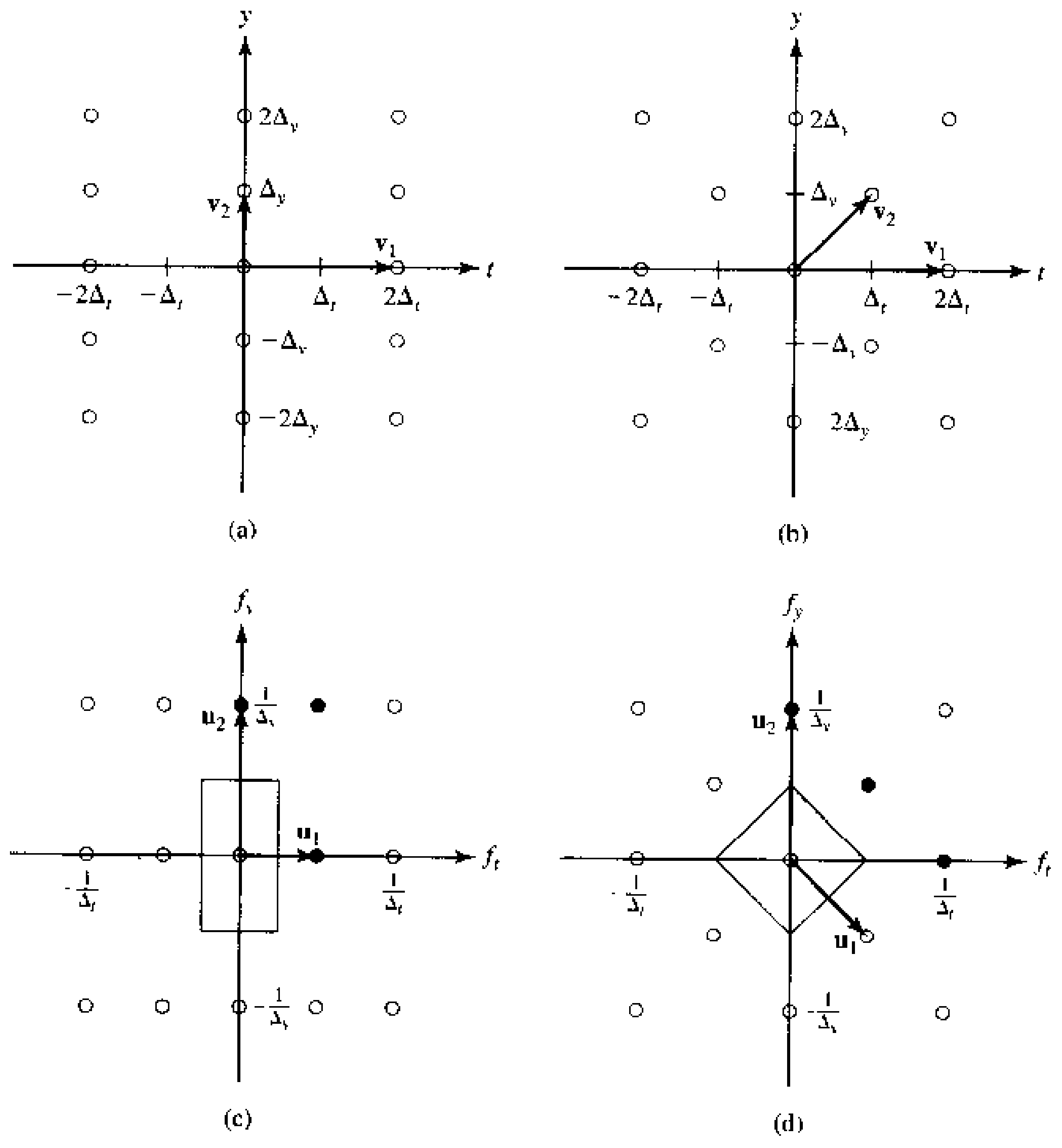


图3.6 逐行和隔行扫描的比较：(a)逐行扫描的采样点阵；(b)隔行扫描的采样点阵；(c)逐行扫描的反商阵；(d)隔行扫描的反商阵。(c)和(d)中的实心圆圈表示最近的混叠分量

比较这两种扫描的原始和反商点阵，我们得到以下几个观点：

1.它们具有相同的采样密度，即$d(\Lambda_1)=d(\Lambda_2)=1/2\Delta_t\Delta_y$。

2.沿垂直频率轴，它们在$1/\Delta_y$具有相同的最近的混叠。这意味着在没有运动时，两种采

样点阵有相同的分辨率。这是因为尽管每场只有一半数目的行,当成像景物静止时,在两个分离的场中采样的行看起来好像在同时被采样一样。然而,当景物存在运动时,隔行扫描的垂直分辨率要低于逐行扫描的垂直分辨率。

3.沿时间频率轴它们具有不同的最近的混叠。对于逐行扫描,第一个混叠出现在 $1/2\Delta_t$ 处,而隔行扫描出现在 $1/\Delta_t$ 处。我们记得,大于最近的时间混叠一半的时间频率成分将会产生闪烁效应。可以看出,当物体具有平坦的或缓慢变化的垂直图案时,隔行扫描的闪烁效应不大可能会发生。

4.它们具有不同的混合混叠。混合混叠定义为最近的偏离轴的混叠分量。接近混合混叠的频率分量引起行间闪烁和爬行。对于逐行扫描,混合混叠出现在$(1/2\Delta_t, 1/\Delta_y)$,而对于隔行扫描出现在$(1/2\Delta_t, 1/2\Delta_y)$。因为隔行扫描的混合混叠更靠近原点,行间闪烁和爬行在隔行扫描中更容易觉察——这就是所谓的隔行效应。

5.对于一个具有各向同性频谱支撑的信号,隔行扫描更有效。特别是,逐行扫描所能表示的无混叠信号频谱的最大半径是 $1/4\Delta_t$,而隔行扫描增加到 $1/2\sqrt{2}\Delta_t$。在说明这个结果时重要的是要记住,这是基于我们把空间和时间频率等同看待的。

注意前面是对具有相同的总采样率的逐行和隔行扫描的比较。在这种情况下,隔行扫描比逐行扫描具有一些优势。为了进一步提高质量,隔行扫描可以被去隔行,以产生一个两倍于总采样率的逐行扫描(我们将在4.2.1节中考虑去隔行问题)。

Beuker 和 Shah 给出了在恒定线性运动下一个隔行帧的有趣的频谱分析[1]。基于这个分析,他们考虑了去隔行技术和其他速率转换问题。

3.3.3 光栅扫描采样:回到 BT.601 格式

如本章开始所讲的,获得完全数字视频信号的一种方法是采样光栅扫描。我们记得,光栅是由相继帧(或场,对于隔行光栅)内的相继水平扫描行构成的一维信号。因此,沿着扫描线的采样间隔直接决定了水平采样间隔。为确定采样间隔,有几个因素必须考虑到。首先,所产生的水平采样间距要与扫描线间的垂直间距匹配,以便使水平和垂直方向的采样频率相近。其次,所产生的三维空间样点应该符合一个期望的采样点阵。例如,若要使每帧或场的样点形成一个矩形栅格,则采样间隔应该能整除行间隔。在1.5.2节所述的 BT.601 数字视频格式的设计中使用了这些准则。按照公式(1.5.2)选择采样间隔,在相继水平扫描行上的采样点垂直对准,并且水平和垂直采样间距近似相等。

前面的讨论是假设视频光栅只有一个亮度分量。对于具有一个亮度和两个色度分量的彩色视频光栅,一个直接的办法是对所有的分量使用相同的采样频率。这将导致 BT.601 4:4:4 格式,如前面图1.8(a)所示。然而,众所周知,HVS 对于亮度信息比色度更敏感。因此,聪明的做法是在比亮度低的分辨率下采样色度分量。在理想情况下,在所有三维上的采样频率应该同等降低。然而,因为光栅是水平扫描行的堆积,所以降低光栅采样频率只降低了水平采样率;而垂直和时间采样率不受影响。例如,如果色度分量的采样率 $f_{s,c}$ 只有亮度分量采样率 f_s 的一半,时间和垂直采样间隔对于色度和亮度分量一样,只是色度分量的水平采样间隔是亮度分量水平采样间隔的二倍。实际上,这是图1.8(b)所示的用于 BT.601 4:2:2 格式的彩色亚采样格式。如果我们沿着扫描光栅对色度分量进行4倍的向下采样,我们将得到图1.8(c)所示

的 BT.601 4:1:1 格式。给定一个 4:2:2 信号，我们可以对垂直维进一步进行两倍的向下采样，这将导致图 1.8(d)所示的 BT.601 4:2:0 格式。注意不能由光栅信号的均匀采样直接获得 4:2:0格式。BT.601 4:2:2 和 4:2:0 信号间的转换将在第 4 章中考虑(习题 4.6～4.7)。

4:1:1 和 4:2:0 格式具有相同的总采样率(样点/秒)，但 4:2:0 格式可以更精确地表示视频信号，因为它以相同的(亮度分量一半)分辨率捕获色度分量中的水平和垂直细节。另一方面，对于 4:1:1 格式，用与亮度分量相同的分辨率表示垂直细节，而只用 1/4 的亮度分辨率表示水平细节。

前面的讨论假设光栅信号的三个色分量是分离的。对于复合彩色视频的采样，必须首先分离出各个彩色分量，然后进行所讨论过的采样过程。

3.3.4 三维视频采样

在上一节讨论的采样方案中，水平样点在所有场中垂直对准。这样的采样方案不能做到最优化。我们也可以在 x-y 平面上以交错的或更复杂的图案采样。更一般地讲，我们可以用一个期望的点阵直接在三维空间采样。制造一个能够实现复杂的三维采样结构的摄像机是很困难的。然而，我们可以首先用一个密集但结构简单的点阵获得样点，然后把它向下转变为期望的点阵。本节将对已用于视频采样的几种点阵的效率进行比较，这里引用的材料来自[4]。为便于表述，我们将假设频率轴 f_x, f_y, f_t 是分别用信号的最大频率 $1/f_{x,\max}, 1/f_{y,\max}, 1/f_{t,\max}$ 定标的。水平、垂直和时间轴 x, y, t 相应地由 $f_{x,\max} f_{y,\max} f_{t,\max}$ 定标。在下面的讨论中采用定标后的无单位变量。为了评价每一种点阵的采样效率，我们进一步假设信号的支撑区是在定标的频率空间中半径为 1 的球。

首先，让我们考虑帧间隔为 Δ_t，行间隔为 Δ_y 的逐行扫描光栅的采样。如果样点以水平间隔垂直对准，那么等效的三维采样点阵是简单的立方体或正交晶体(ORT)，如图 3.7(a)所示。显然，为了避免混叠，我们必须取 $\Delta_x = \Delta_y = \Delta_t = \frac{1}{2}$，采样密度 $d(\text{ORT}) = 8$。

然后，让我们考虑间隔为 $\Delta_t/2$，每场行间隔为 $2\Delta_y$ 的隔行扫描光栅的采样。如果不同场的样点以水平间隔 Δ_x 对准，那么等效的三维采样点阵如图 3.7(b)所示，称为 ALI。可以证明，为了避免混叠，我们必须取 $\Delta_x = \Delta_y = \frac{1}{2}, \Delta_t = \frac{1}{\sqrt{3}}$，采样密度为 $d(\text{ALI}) = 4\sqrt{3}$。这比 ORT 点阵效率更高。

为了提高采样效率，我们可以使偶行与奇行的样点错开一半，如图 3.7(c)所示。其结果称为体心正交晶体(BCO)。可以证明，为了避免混叠，所需的采样间隔为 $\Delta_x = \Delta_t = \frac{1}{\sqrt{2}}, \Delta_y = \frac{1}{2\sqrt{2}}$，采样密度为 $4\sqrt{2}$。因此这种点阵比 ALI 效率更高。

BCO 采样点阵的一种变形称为面心正交晶体(FCO)，如图 3.7(d)所示。对于这种采样结构，每一场包含所有的行，但同一场的样点在奇、偶行交错。这种点阵的采样效率与 BCO 一样，但由于奇、偶场的样点被更均匀地混合，通常导致更好的视觉质量。

这些点阵的最小采样密度和相应的采样间隔概括在表 3.1 中。

表 3.1 具有单位球频谱的信号的无混叠采样点阵

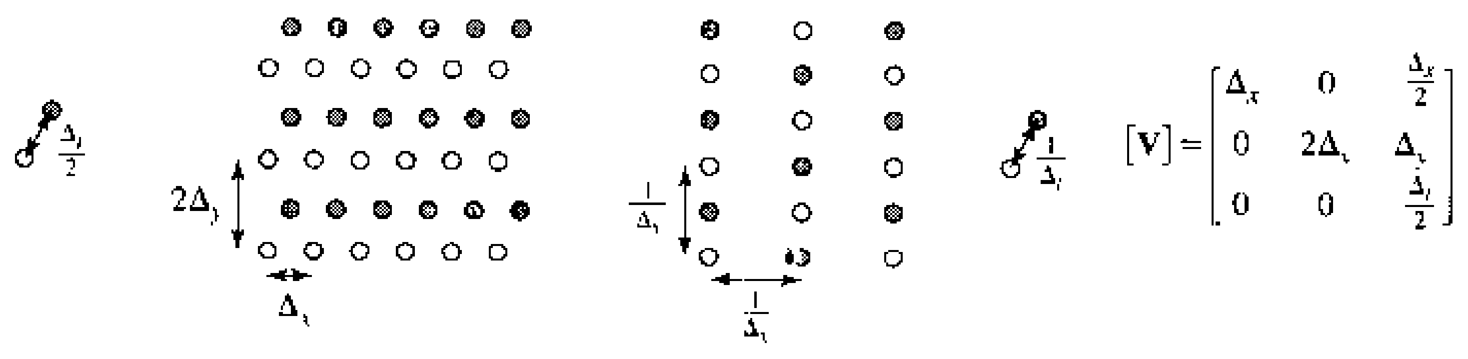
点阵	Δ_x	Δ_y	Δ_t	密度	效率
ORT	1/2	1/2	1/2	8	$\pi/6 = 0.524$
ALI	1/2	1/2	$1/\sqrt{3}$	$4\sqrt{3}$	$\pi/3\sqrt{3} = 0.605$
BCO	$1/\sqrt{2}$	$1/2\sqrt{2}$	$1/\sqrt{2}$	$4\sqrt{2}$	$\pi/3\sqrt{2} = 0.740$
FCO	1/2	1/2	$1/\sqrt{2}$	$4\sqrt{2}$	$\pi/3\sqrt{2} = 0.740$

$$[\mathbf{V}] = \begin{bmatrix} \Delta_x & 0 & 0 \\ 0 & \Delta_y & 0 \\ 0 & 0 & \Delta_t \end{bmatrix}$$

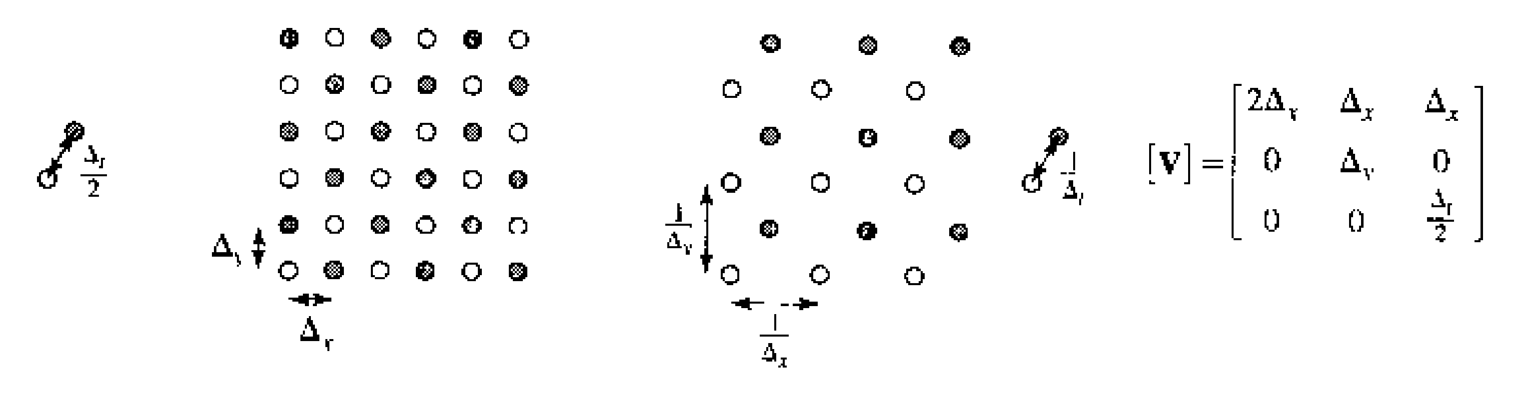
(a) 正交晶体点阵(ORT)

$$[\mathbf{V}] = \begin{bmatrix} \Delta_x & 0 & 0 \\ 0 & 2\Delta_y & \Delta_y \\ 0 & 0 & \frac{\Delta_t}{2} \end{bmatrix}$$

(b) 2:1隔行信号垂直对准采样获得的点阵(ALI)



(c) 体心正交晶体点阵(BCO)



(d) 面心正交晶体点阵(FCO)

图 3.7 几种三维视频采样点阵和它们的反商阵。在图中的每一部分，左图是采样点阵，右图是反商点阵。矩阵[V]代表该采样点阵的生成矩阵。空心和实心圆圈代表两个不同的时间(左图)或时间频率(右图)平面内的阵点，其间的距离分别用左右两边的箭头表示。例如，对于(b)，左图中空心圆圈代表在 $t = 0$ 时的采样，实心圆圈代表在 $t = \Delta_t/2$ 时的采样。右图中空心圆圈代表在时间频率 f_t 时的混叠分量，实心圆圈代表在 $f_t = 1/\Delta_t$ 时的混叠分量。对于(a)，实心圆圈正好落在空心圆圈的后面，因此没有表示出来

3.3.5 空间和时间混叠

在 2.3.2 节描述了时间频率、空间频率与进行线性运动的物体的速度之间的关系。一般地说，如果物体的速度较高（除非它的运动方向与空间频率的方向正交），那么它的时间频率就较高。图 3.8 说明时间混叠的效果，还是考虑线性运动的物体。

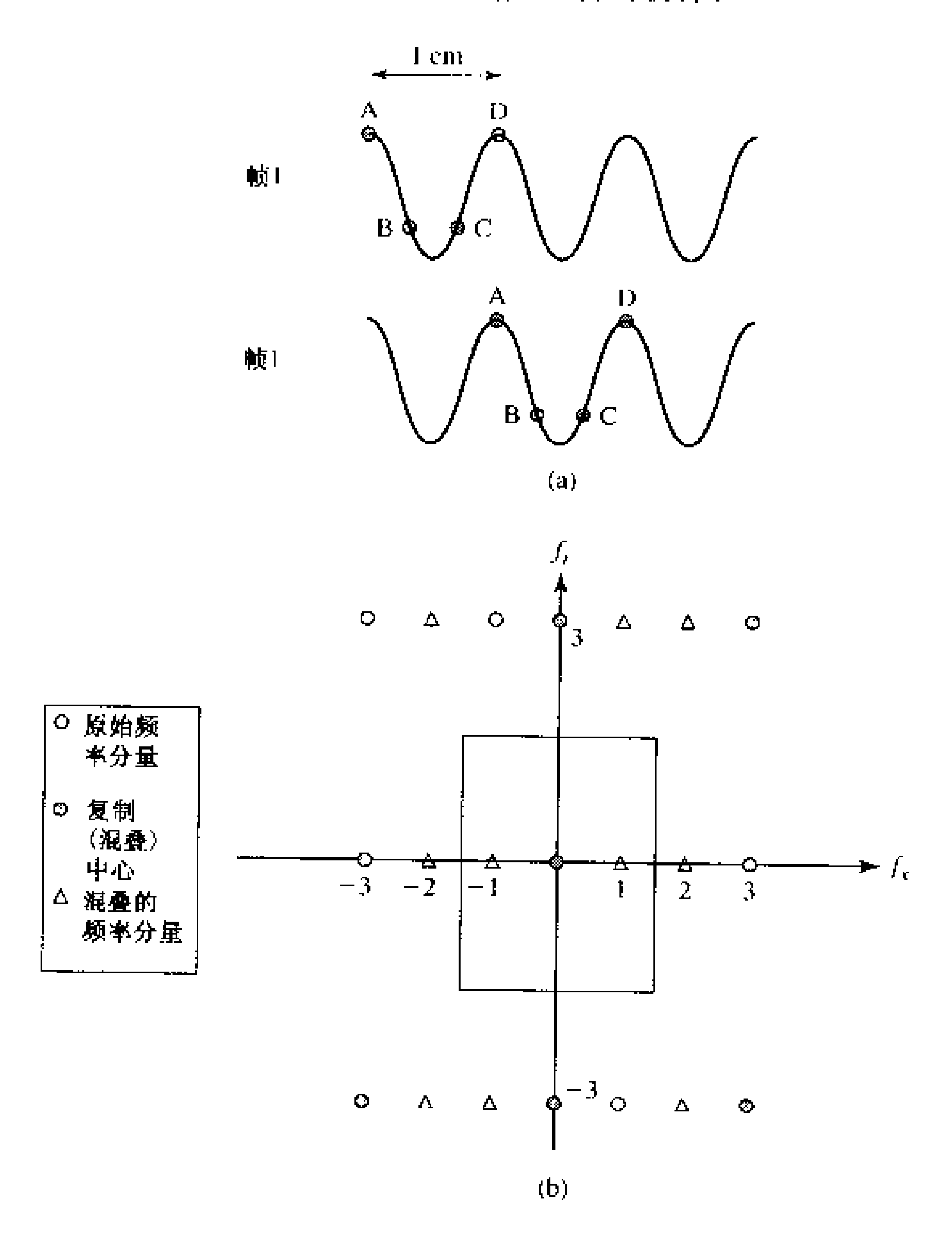


图 3.8 正弦图案的采样

例 3.4 考虑一个具有纯垂直正弦彩色图案、水平频率为 1 周期/厘米的景物（见图 2.1(a)）。假设这个图案以 $v_x = 3$ 厘米/秒的速度水平运动。如果我们用 $f_{s,x} = f_{s,y} = 3$ 样点/厘米，$f_{s,t} = 3$ fps 的速率对它进行采样，那么表观运动和正弦频率是多少？

因为信号的空间频率为 1 周期/厘米，并且以 3 厘米/秒的速度运动，在时间采样率为 3 fps时，它超前 3 cps 或 1 周期/帧。因此在两个相继的帧所捕获的帧是相同的，如图 3.8(a)所示。因为空间采样率足够高（$f_{s,x} > 2f_{x,0}$），所以没有空间混叠。因此，采样信号将表现为一个静态的正弦，具有与连续信号一样的图案。

现在用点阵采样定理解释这个现象。显然这个信号在$(f_{x,0}, f_{y,0}) = \{(1,0), (-1,0)\}$

处有一对空间频率分量。基于公式(2.3.4),由运动产生的时间频率为 $f_{t,0} = -(f_{x,0}v_x + f_{y,0}v_y) = \{-3,3\}$(cps)。因此,信号频谱在$(f_{x,0},f_{y,0},f_{t,0}) = \{(1,0,-3),(-1,0,3)\}$处有一对脉冲,如图3.8(b)所示(在这个图中只画了在 $f_y=0$ 时的 f_x-f_t 平面,因为信号在所有其他平面上为零)。采样后,信号频谱将在所有采样频率的整数倍处复制,如图所示。我们可以看到,因为 $f_{s,t} < 2f_{t,0}$,一对混叠分量落在反商阵的沃罗纳晶格内,在这种情况下它是以 $\pm f_{s,x}/2$ 和 $\pm f_{s,t}/2$ 为边界的矩形区域。实际上原始信号分量落在这个区域之外。如果用理想的低通滤波器由采样信号恢复原始信号,那么恢复的信号将是空间频率为(1,0),时间频率为0的纯正弦。它相当于一个水平频率为1周期/厘米的静止垂直正弦。这就是将被眼睛感知的信号,眼睛的作用类似于一个低通滤波器。

为了在采样信号中保留真实的运动,时间采样率应该至少为 $2f_{t,0} = 6$ fps。在这个速率下,每一帧是前一帧移动半个周期。

例3.4代表了时间混叠效应的极限情况:运动物体变为静止。更一般地说,当帧率不够高时,快速运动的物体会看起来以较低的速度运动。取决于运动物体的空间图案,一些线状图形的运动方向也可能被改变。

3.4 摄像机和显示器的滤波作用

前面的几节介绍了一般 K 维信号和视频信号的采样定理。对于几种典型的视频格式,我们已经导出了所需要的预滤波器和重建滤波器。在本节中,我们讨论实际的摄像机和显示器是如何以一种粗略的方式完成这些任务的,以及 HVS 是如何部分地完成内插任务的。

3.4.1 摄像机孔径

考虑一个在水平、垂直和时间方向上分别以常规间隔 $\Delta_x,\Delta_y,\Delta_t$ 采样相继变换场景的摄像机。这相当于使用简单的立方体点阵。采样频率是 $f_{s,x}=1/\Delta_x, f_{s,y}=1/\Delta_y, f_{s,t}=1/\Delta_t$。理想的预滤波器应该是截止频率为采样频率一半的低通滤波器。下面讨论在典型的摄像机中实际的预滤波器的实现。

时间孔径 视频摄像机一般是在摄取过程中完成一定程度的预滤波的。首先,在任何一帧期间读出的亮度值不是当时的感光值;而是一个时间间隔 Δ_e 内感光信号的平均值,这个时间间隔称为曝光时间。因此,摄像机在时域应用了一个脉冲响应为如下形式的预滤波器:

$$h_{p,t}(t) = \begin{cases} \dfrac{1}{\Delta_e}, & t \in (0,\Delta_e) \\ 0, & \text{其他} \end{cases} \tag{3.4.1}$$

这个滤波器的频率响应为:

$$H_{p,t}(f_t) = \exp(-j\pi f_t\Delta_e)\frac{\sin(\pi f_t\Delta_e)}{\pi f_t\Delta_e} \tag{3.4.2}$$

我们可以看到它在 $f_t = 1/\Delta_e$ 处达到0。回忆 $1/\Delta_t$ 是时间采样率,而理想的预滤波器是截止频率为采样频率一半的低通滤波器,当选择 $\Delta_e \geq \Delta_t$ 时,摄像机可以抑制采样率附近的时间混叠分量。但 Δ_e 太大将会使信号模糊。实际上,模糊效应有时比混叠更为明显。因此曝光时间

Δ_t 的选择必须在混叠和模糊效应之间达到一个适当的折中。

空间孔径 除了时间积分作用以外，摄像机还具有空间积分作用。在任何一个像素(电子管摄像机的一个扫描行上的一个位置或CCD摄像机的一个传感器)读出的值不只是那个点的光信号，而是在围绕它的一个小窗口内信号的加权积分，这一小窗口称为孔径。孔径的形状和加权值构成了摄像机的空间孔径函数。这个孔径函数起着空间预滤波器的作用，它的傅里叶变换称为摄像机的调制转移函数(MTF)。对于大多数摄像机，空间孔径函数可以近似为圆对称高斯函数：

$$h_{p,x,y}(x,y) = \frac{1}{\sqrt{2\pi}\sigma}\exp(-(x^2+y^2)/2\sigma^2) \tag{3.4.3}$$

这个函数的频谱也是高斯函数：

$$H_{p,x,y}(f_x,f_y) = \exp(-(f_x^2+f_y^2)/2\beta^2),\beta = \frac{1}{2\sigma\pi} \tag{3.4.4}$$

σ 或 β 的值依赖于孔径的大小和形状，通常设计成使得在水平和垂直采样率一半处的频率响应为0.5。假设 $f_{s,x} = f_{s,y} = f_{s,x,y}$，我们看到这需要 $\beta = f_{s,x,y}/2\sqrt{\ln 2}$。关于几种摄像管的MTF可参见[6]。

联合孔径 摄像机总的孔径函数或预滤波器是：

$$h_p(x,y,t) = h_{p,t}(t)h_{p,x,y}(x,y) \tag{3.4.5}$$

频率响应为：

$$H_p(f_x,f_y,f_t) = H_{p,t}(f_t)H_{p,x,y}(f_x,f_y) \tag{3.4.6}$$

一个 $\Delta_e = \Delta_t = 1/60$ 秒和 $f_{s,y} = 480$(行/像高)的摄像机的脉冲响应如图3.9(a)所示。它的频率响应在图3.9(b)中给出。只示出了 $f_x = 0$ 时的 $f_y - f_t$ 平面。显然，它与理想的半带低通滤波器相去甚远，理想情况下它应该是由 $|f_y| \leq f_{s,y}/2, |f_t| \leq f_{s,t}/2$ 定义的方形通带。一方面，它衰减了期望通带(沃罗纳晶格)内的频率分量，因此不必要地降低了信号的分辨率；另一方面，它没有完全除去在期望阻带内的频率分量，将导致采样信号的混叠。已经发现，观察者对分辨率的降低比混叠效应更难于忍受。部分原因是，只当图像含有与最低混叠频率接近的高频周期图案时，混叠效应才引起可觉察的视觉效应，而这种情况在自然景物图像中是少见的。由于这个原因，保持通带内的信号比抑制通带外的信号更为重要。

为了实现更精确的预滤波，可以采用数字滤波器。这包括三个步骤：(1)以高于期望采样率的速率采样信号；(2)用数字滤波器抑制期望通带外的频率分量；(3)把数字信号向下转换到期望的速率。尽管具有很尖锐过渡的滤波器在均方误差方面能给出较好的结果，但这种滤波器在陡峭边缘处会产生振铃，称为吉布斯(Gibbs)效应。而且，要实现尖锐过渡，需要很高阶的滤波器，这在视频应用中可能是不可行的。

3.4.2 显示器孔径

如1.2节所述，在CRT监视器中，电子枪在屏幕上一行行发射电子束，以与视频信号的强度成比例的强度轰击相应位置的荧光体。为了显示彩色图像，用三个分离的电子枪发射三个电子束，以期望的强度组合轰击每个位置上的红、绿和蓝三个荧光体。电子束的粗细本质上决定了垂直滤波：很细的束将使图像看起来更清楚，但如果观察者离屏幕太近时，也会看到扫描线；另一方面，粗的束将使图像模糊。通常，为了使空间分辨率的损失最小化而使用细的电子

束,这样显示器的垂直滤波作用很少。时间滤波是由荧光体确定的。用于彩色电视机的 P22 荧光体在 10 μs 到 1 ms 内衰减到峰值响应的 10%以下[3,4],远小于场时间(16.7 ms)。因此实际上没有进行时间滤波。

幸运的是,HVS 具有低通或带通特性,依赖于图像的时间和空间频率的制式(如 2.4 节所述)。因此,在一定程度上眼睛完成了所需的内插任务。为了改善性能,我们可以使用数字滤波器把采样信号向上转换为更高的分辨率,然后把它送到一个高清晰度显示系统。

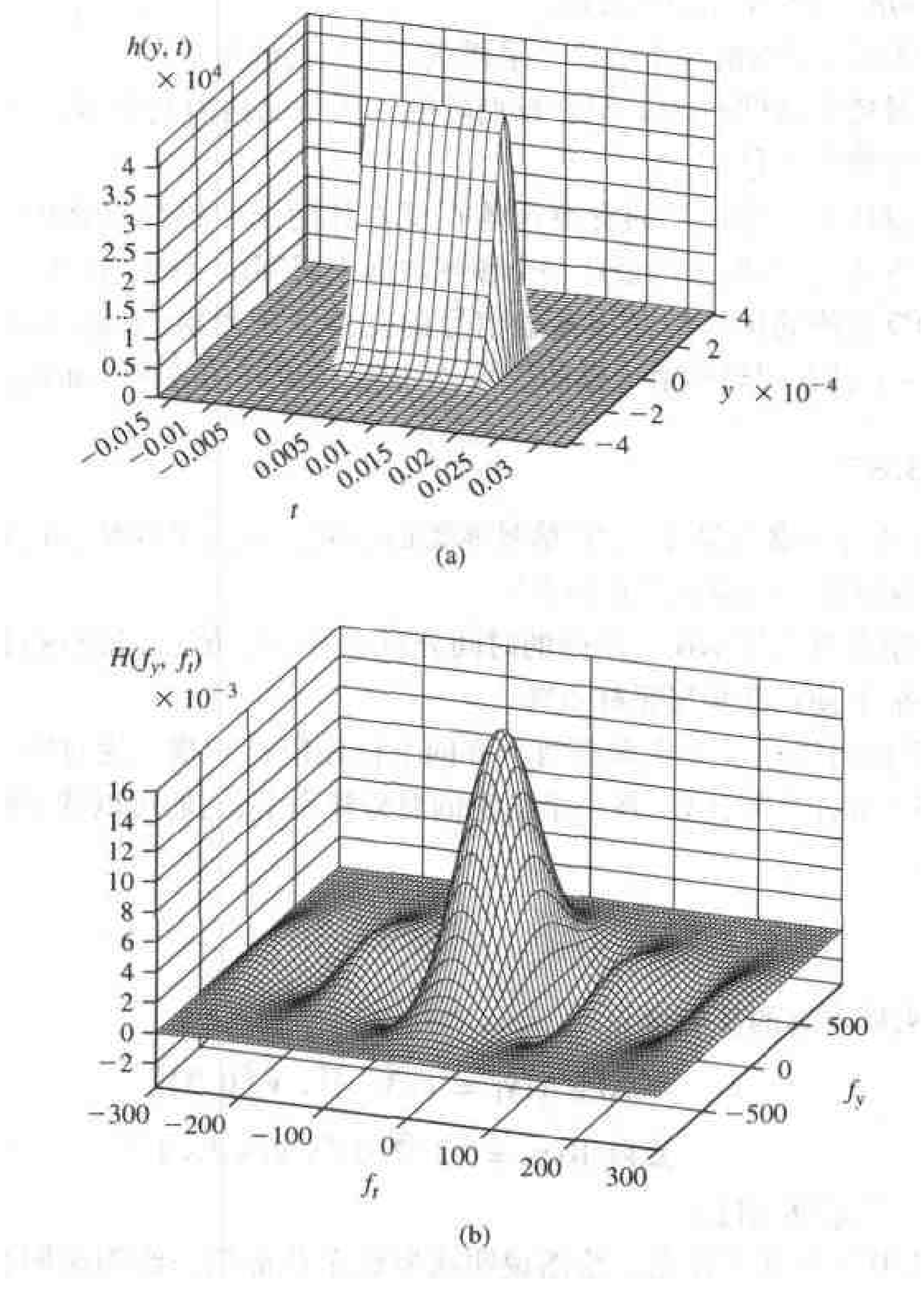


图 3.9 一个典型摄像机的孔径函数:(a) 脉冲响应,(b) 频率响应

已经发现,在大多数现行电视系统中,摄像机和显示器孔径的联合作用导致垂直最大分辨率只是理论极限(行率的一半)的 0.7。换句话说,一个行数为 N_y 的电视系统所能恰当表示的最大垂直频率大约为 $0.7N_y/2$。这个因子称为凯尔因子[4],记为 K。关于一个成像系统的参数如何影响凯尔因子的讨论,读者可以参阅参考文献[8]。使用数字滤波器进行预滤波和内插,有可能使凯尔因子非常接近 1。在 1.4.3 节中,我们已经看到凯尔因子是如何影响电视信号带宽的。

3.5 小结

在任意点阵上采样 K 维信号(3.2 节)

- 一个 K 维连续空间信号可以在一个点阵上被采样。我们可以定义采样空间傅里叶变换(SSFT),它满足与 DSFT 相似的性质。
- SSFT 是周期性的,它的混叠中心位于采样点阵的反商阵点上。
- 如果原始信号的频谱(即 CSFT)只在反商阵的沃罗纳晶格内是非零的,那么可以由采样信号完全恢复原始信号。
- 采样点阵的设计应该使信号的支撑区落在反商阵的沃罗纳晶格之内。否则会产生混叠。另一种方法是,我们可以设计一个预滤波器来限制信号的支撑区。
- 采样点阵的反商阵的沃罗纳晶格覆盖信号的支撑区越紧凑,采样点阵的效率就越高。例如,对于一个具有圆形频谱支撑的信号,矩形采样点阵的效率不如六边形点阵。

视频信号的采样(3.3 节)

- 除依赖于信号的频谱分量外,视频信号所需的空间和时间采样频率还依赖于视觉阈值(即 HVS 的空时频率响应的截止频率)。
- 隔行扫描是用垂直分辨率换取增强的时间分辨率的一种方法。隔行扫描可以理解为在垂直时间平面上使用非矩形采样点阵。
- 更一般地,我们可以在水平和垂直两个方向上使用隔行扫描。这对应于三维的非立方体点阵。与立方体点阵比较,在一个给定的总采样率下,它们可以减少混叠效应。

3.6 习题

3.1 设两个采样点阵的基矢量为:

$$\text{点阵 A}: \mathbf{v}_1 = [\sqrt{3},1]^T,\ \mathbf{v}_2[0,2]^T$$

$$\text{点阵 B}: \mathbf{v}_1 = [2\sqrt{2},0]^T,\ \mathbf{v}_2[\sqrt{2},\sqrt{2}]^T$$

对于每一个采样点阵:

(a) 画出基矢量和采样点。绘图说明沃罗纳单位晶格。绘图说明整个空间域是如何由该单位晶格的移位副本铺成的。确定采样密度。

(b) 确定反商点阵的基矢量。对于这个点阵重复(a)。

(c) 对于一个具有圆形频谱的信号(即支撑区为一个圆),哪一种点阵更好?构造一个频谱(画出它的支撑区),它用某个点阵将产生混叠,而用其他点阵则不产生混叠。

(d) 对于一个给定的点阵,基矢量不是惟一的,你能为这些点阵找出另一个基矢量集吗?

3.2 证明定理 3.2。

3.3 证明定理 3.3。

3.4 证明定理 3.5。

3.5 考虑一个在连续空间$\mathcal{R}^K$ 上的脉冲响应为 $h_c(\mathbf{x})$的线性移不变系统。请说明：如果在点阵 Λ 上对输入输出信号采样，采样的输入信号$\psi_s(\mathbf{n}) = \psi_c([\mathbf{V}]\mathbf{n})$和输出信号 $\phi_s(\mathbf{n}) = \phi_c([\mathbf{V}]\mathbf{n})$由一个 Λ 内的采样空间卷积联系起来，并且滤波器正好是采样脉冲响应 $h_s(\mathbf{n}) = h_c([\mathbf{V}]\mathbf{n})$。

3.6 在 3.2.5 节中，我们说明了在单位超立方体$\mathcal{I}^K$ 内的 DSFT 是在(用变换$[\mathbf{U}]\mathbf{f}$)使$\mathcal{I}$扭曲所得的区域内的 SSFT 的扭曲图像。当采样点阵是六边形时，绘图说明这个扭曲过程。更特殊一些，考虑一个具有球形支撑区的信号。从信号在六边形点阵上采样开始，画出信号的 SSFT 和 DSFT。

3.7 考虑一个如下定义的二维信号：

$$\psi(x,y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2+y^2}{2}} \tag{3.6.1}$$

(a) 确定它的 CSFT。

(b) 假设我们想用由下式给出的六边形点阵 Λ 采样这个信号：

$$[\mathbf{V}] = \alpha \begin{bmatrix} \sqrt{3}/2 & 0 \\ 1/2 & 1 \end{bmatrix} \tag{3.6.2}$$

选择一个适当的比例因子 α 使得混叠效应不严重。例如，选取 α 使在 Λ^* 的沃罗纳晶格的边界处$\Psi(f_x,f_y) = 0.1$。

(c) 确定采样信号和它的频谱 SSFT。

(d) 用 MATLAB 画出原始连续信号、采样信号及连续的和采样的频谱。在 SSFT 中你能看到重复的峰值吗？

(e) 确定并画出采样信号的 DSFT，并与 SSFT 比较。DSFT 和 SSFT 是由公式(3.2.16)表示的扭曲关系联系起来的吗？

3.8 对于图 3.7 所示的三维采样点阵，证明为了避免混叠，如果信号的支撑区是单位球，那么采样间隔应按表 3.1 所示来选取。

3.9 考虑一个由下式表述的正弦条图案：

$$\psi(x,y) = \sin(4\pi(x-y))$$

设 x 和 y 方向的单位是米(m)。假设这个图形以(v_x,v_y)m/s 的速度运动。对于以下速度确定垂直、水平和时间方向所需要的采样率：

(a) $(v_x,v_y) = (1,1)$

(b) $(v_x,v_y) = (-1,1)$

(c) $(v_x,v_y) = (2,1)$

3.10 (继续习题 3.9)。假设正弦条以速度$(v_x,v_y) = (3,0)$运动。如果信号的采样率为 $f_{s,x} = f_{s,y} = 6$ 样点/米，$f_{s,t} = 6$ 样点/秒，那么采样的条图案的表观空间频率、方向和速度是多少？

3.11 考虑一个做如下恒定线性运动的一个场景：

$$\psi(x,y,t) = \psi(x - v_x t, y - v_y t, 0) = \psi_0(x - v_x t, y - v_y t) \tag{3.6.3}$$

如果用一个隔行点阵采样这个连续图像以形成奇场和偶场，那么在时间 t 偶场的 DSFT 与 $\psi_0(x,y,t)$ 的 CSFT 有何关系？奇场的 DSFT 和隔行帧的 DSFT 有何关系？(参见[1])。

3.12 在 3.4.1 节中，我们考虑了一个具有高斯形状的空间孔径函数的摄像机。请说明它的频谱确实如公式(3.4.4)所给出的那样吗？当频率响应在 $f_x = f_{s,x}/2, f_y = f_{s,y}/2$ 处为 0.5 时，β 值应该为多少？

3.13 对于某些实际的摄像机，它们的空间孔径函数可以用一个方盒函数较好地模型化：

$$h_{p,x,y}(x,y) = \begin{cases} \frac{1}{T_x T_y} & |x| < T_x/2, |y| < T_y/2 \\ 0 & \text{其他} \end{cases} \tag{3.6.4}$$

假设时间脉冲响应与公式(3.4.1)所给出的相同。试求摄像机的总孔径函数 $h_p(x,y,t)$ 的 CSFT。假设 $T_x = \Delta_x = 1/720$ 像宽，$T_y = \Delta_y = 1/480$ 像高，$\Delta_e = \Delta_t = 1/60$ 秒，画出其幅度响应。将这个频谱与具有高斯孔径(见图 3.9)的摄像机比较，哪一个的预滤波效率更高？

3.7 文献目录

[1] Beuker, R. A., and I. A. Shah. Analysis of interlaced video signals and its applications. *IEEE Trans. Image Process.* (Sept. 1994), 3(5):501–12.

[2] Cassels, J. W. S. *An Introduction to the Geometry of Numbers*. Berlin: Springer-Verlag, 1959.

[3] Diakides, N. A. Phosphor screens. In D. G. Fink, ed., *Electronics Engineers' Handbook*. New York: McGraw Hill, 1975, 11-33–39.

[4] Dubois, E. The sampling and reconstruction of time-varying imagery with application in video systems. *IEEE* (1985), 73:502–22.

[5] Dudgeon, D. E., and R. M. Mersereau. *Multidimensional Digital Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1984.

[6] Miller, L. D. A new method of specifying the resolving power of television camera tubes using the RCA P-300 test chart. *Journal of Society of Motion Picture Television Engineering*, (Apr. 1980), 89:249–56.

[7] Petersen, D. P., and D. Middleton. Sampling and reconstruction of wave-number-limited functions in *N*-dimensional Euclidean spaces. *Information Control* (1962), 5:279–323.

[8] Tonge, G. J. The television scanning process. *Journal of Society of Motion Picture Television Engineers*, (July 1984), 93:657–66.

第4章　视频采样率转换

在第3章中我们考虑了采样问题，它是将原本连续的视频信号转换为数字信号时所必需的步骤。在数字域中，我们经常需要把数字视频信号从一种格式（指空间和时间的分辨率）转换为另一种格式。例如将以PAL格式记录的视频转换为NTSC格式，从隔行扫描转换为逐行扫描，从HDTV格式的信号转换为SDTV格式，等等。本章将考虑采样率转换问题。4.1节讲述在不同点阵上采样的多维信号转换的一般理论。4.2节讲述三维视频信号采样率转换问题，并且以前面提到的视频转换问题为例进行讨论。

4.1　在不同点阵上采样信号的转换

给定一个定义于点阵 Λ_1 上的已采样信号，经常需要产生一个定义于另一点阵 Λ_2 上的信号。这就构成了采样率转换问题。解决的方法取决于这两个点阵之间的关系。例如，如果 $\Lambda_1 \subset \Lambda_2$，也即，$\Lambda_1$ 中的每一个点也在 Λ_2 中，那么此问题是上转换（或内插）问题。我们可以先将那些在 Λ_2 中而不在 Λ_1 中的点填零（即零填充），然后可以用一个作用于 Λ_2 上的内插滤波器估计这些点的值。在4.1.1节中将讨论这种内插滤波器所要求的形式。

另一方面，若 $\Lambda_1 \supset \Lambda_2$，即为下转换（或抽取）问题。我们可以简单地从 Λ_1 中取出那些也在 Λ_2 中的点。然而，为避免下采样信号中出现混叠，我们需要对信号进行预滤波，以将其带宽限制到 Λ_2^* 的沃罗纳晶格。上转换和下转换的过程示于图4.1(a)和图4.1(b)。

一般，如果 Λ_1 和 Λ_2 互相不包含，就需要找到另一个既包含 Λ_1 又包含 Λ_2 的点阵 Λ_3。我们可以首先将 Λ_1 上采样到 Λ_3，然后再将 Λ_3 下采样到 Λ_2。此过程示于图4.1(c)。图中，Λ_3 中的中间滤波器完成两个任务：首先，内插出 Λ_1 中漏下的采样点；其次，把 Λ_3 中信号的频谱限制到 Λ_2^* 的沃罗纳晶格。

在下面的小节中将分别讨论上述情况。为简化符号表示，我们用$\mathcal{V}_i^*$ 表示 Λ_i^* 的沃罗纳晶格$\mathcal{V}(\Lambda_i^*)$。

4.1.1　上转换

如前所述，如果 $\Lambda_1 \subset \Lambda_2$，那么我们可以首先把 Λ_1 中的样点传递到 Λ_2 中，漏下的样点用零填充。这个上采样的过程可以描述成如下的形式：

$$\psi_{s,3}(\mathbf{x}) = \mathcal{U}\{\psi_{s,1}\}(\mathbf{x}) = \begin{cases}\psi_{s,1}(\mathbf{x}), & \mathbf{x} \in \Lambda_1 \\ 0, & \mathbf{x} \in \Lambda_2 \setminus \Lambda_1\end{cases} \tag{4.1.1}$$

其中 $\mathbf{x} \in \Lambda_2 \setminus \Lambda_1$ 表示在 Λ_2 中但不在 Λ_1 中的点集。现在此信号定义于点阵 Λ_2 上。

我们需要用一个定义于 Λ_2 上的内插滤波器来将漏下的样点填零。为了确定合适的内插滤波器，让我们返回到定义于连续空间$\mathcal{R}^K$ 中的原始连续信号$\psi_c(\mathbf{x})$。如果在 Λ_2 中用生成矩

阵$[\mathbf{V}_2]$直接采样得到信号，那么，采样信号的 SSFT 将是：

$$\Psi_{s,2}(\mathbf{f}) = d(\Lambda_2)\sum_{\mathbf{m}\in Z^K}\Psi_c(\mathbf{f}-[\mathbf{U}_2]\mathbf{m}) \tag{4.1.2}$$

其中，$[\mathbf{U}_2]=([\mathbf{V}_2]^T)^{-1}$。另一方面，在 Λ_1 中用生成矩阵为$[\mathbf{V}_1]$的采样信号的 SSFT 是：

$$\Psi_{s,1}(\mathbf{f}) = d(\Lambda_1)\sum_{\mathbf{m}\in Z^K}\Psi_c(\mathbf{f}-[\mathbf{U}_1]\mathbf{m}) \tag{4.1.3}$$

其中，$[\mathbf{U}_1]=([\mathbf{V}_1]^T)^{-1}$。如果 Λ_1 满足公式(3.2.9)中的无混叠条件，那么在 Λ_1^* 中的混叠分量之间不会发生重叠。

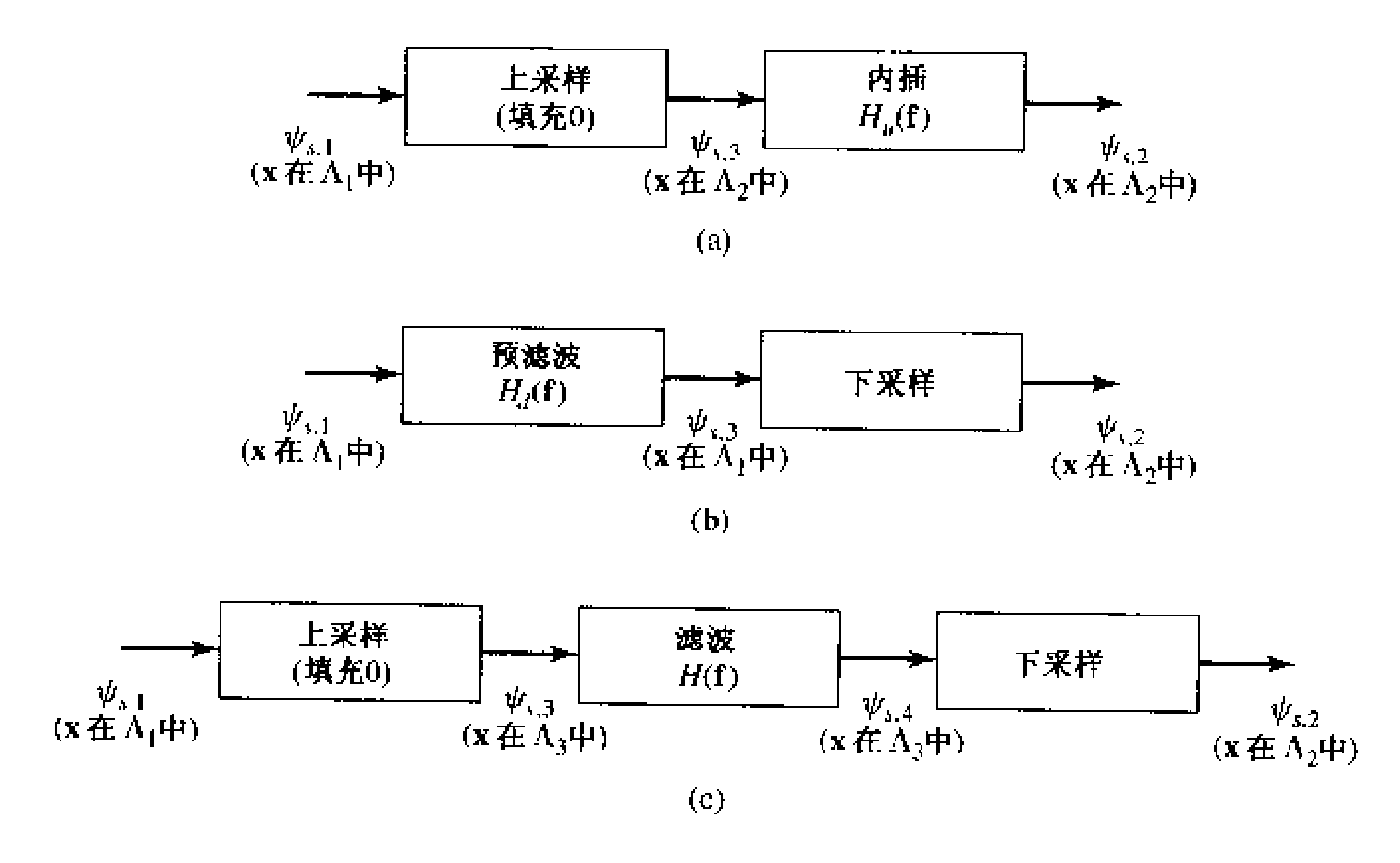


图 4.1 采样率转换：(a)上转换 $\Lambda_1\subset\Lambda_2$；(b) 下转换 $\Lambda_2\subset\Lambda_1$；(c)任意采样率转换 $\Lambda_3=\Lambda_1+\Lambda_2$

由于 $\Lambda_1\subset\Lambda_2$，根据定理 3.3，$\Lambda_1^*\supset\Lambda_2^*$，从而$\mathcal{V}_1^*\subset\mathcal{V}_2^*$。为从$\Psi_{s,1}(\mathbf{f})$中导出$\Psi_{s,2}(\mathbf{f})$，我们只需从$\mathcal{V}_2^*$ 中去掉$\Psi_{s,1}(\mathbf{f})$的不在原始频谱中的那部分。当不能准确地知道原始频谱时，最好的方法是假设它与$\Psi_{s,1}(\mathbf{f})$在其沃罗纳晶格$\mathcal{V}_1^*$ 中的部分相同。因此，在理想情况下，内插滤波器的频率响应为：

$$H_u(\mathbf{f}) = \begin{cases} d(\Lambda_2)/d(\Lambda_1), & \mathbf{f}\in\mathcal{V}_1^* \\ 0, & \mathbf{f}\in\mathcal{V}_2^*\setminus\mathcal{V}_1^* \end{cases} \tag{4.1.4}$$

在上述公式中，$H_u(\mathbf{f})$表示定义于 Λ_2 上的采样空间滤波器的 SSFT。

例 4.1 考虑如图 4.2 所示的从采样点阵 Λ_1 到 Λ_2 的上转换，它们各自的生成矩阵是：

$$[\mathbf{V}_1] = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix};[\mathbf{V}_2] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.1.5}$$

利用公式(3.1.7)，反商点阵的生成矩阵是：

$$[\mathbf{U}_1] = \begin{bmatrix} 1/2 & 0 \\ -1/2 & 1 \end{bmatrix};[\mathbf{U}_2] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{4.1.6}$$

(图 4.2 中也示出了反商点阵和它们的沃罗纳晶格。)假设原始连续信号的频谱具有圆形支撑，在图中也画出了对应于不同采样图案的重复频谱。可以看到，实际上$\mathcal{V}_2^*\supset\mathcal{V}_1^*$。理

想的内插滤波器为：

$$H_u(\mathbf{f}) = \begin{cases} d(\Lambda_2)/d(\Lambda_1) = 2, & \mathbf{f} \in \mathcal{V}_1^* \\ 0, & \mathbf{f} \in \mathcal{V}_2^* \setminus \mathcal{V}_1^* \end{cases} \quad (4.1.7)$$

在图 4.2 中示于点阵 Λ_2^* 上。这个滤波器的作用是消除 Λ_1^* 中不应该在 Λ_2^* 内出现的混叠分量。

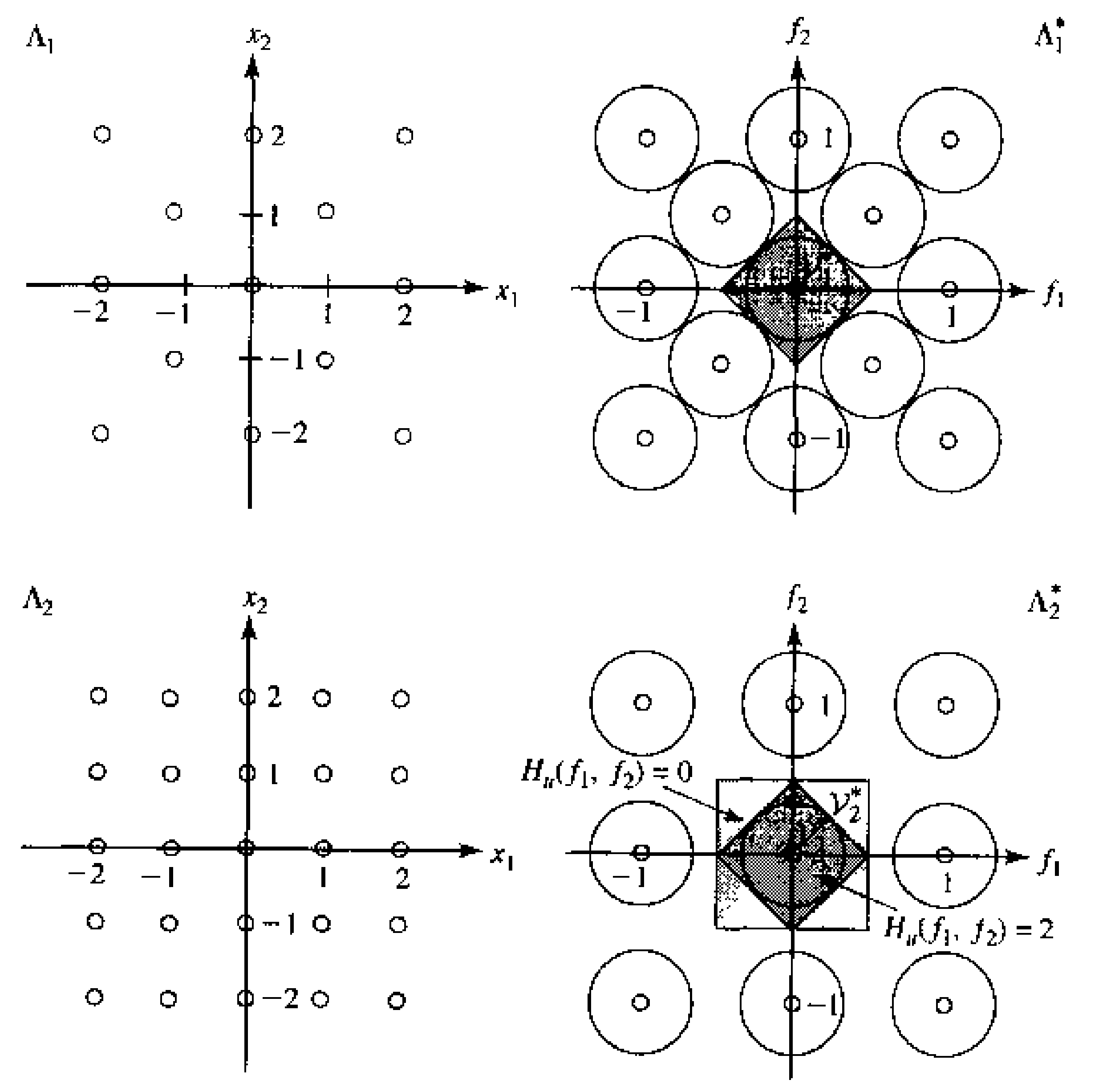


图 4.2　一个从点阵 Λ_1 到 Λ_2 上转换的例子。理想的内插滤波器示于 Λ_2^*（见例 4.1）

4.1.2　下转换

如果 $\Lambda_1 \supset \Lambda_2$，我们可以通过保留既在 Λ_1 中也在 Λ_2 中的样点，丢弃所有其他样点，而容易地得到 $\psi_{s,2}$。这个下采样的过程可以描述为：

$$\psi_{s,2}(\mathbf{x}) = \mathcal{D}\{\psi_{s,1}\}(\mathbf{x}) = \psi_{s,1}(\mathbf{x}), \mathbf{x} \in \Lambda_2 \quad (4.1.8)$$

这样得到的信号将与使用相同的预滤波器在 Λ_1 上对原始连续信号采样所得到的信号完全相同。假定原始连续信号的支撑区或者在获取 $\psi_{s,1}$ 时预滤波后的支撑区等于或小于 $\mathcal{V}_1^*$，使得在 $\psi_{s,1}$ 中不存在混叠。那么，一般地说，由于 $\mathcal{V}_2^*$ 较小，在 $\psi_{s,1}$ 中一定会有混叠。为了避免混叠，需要对 $\psi_{s,1}$ 进行预滤波，以去除在 $\mathcal{V}_1^*$ 中但不在 $\mathcal{V}_2^*$ 中的频率分量。因此，Λ_1 上合适的预滤波器用 Λ_1 上的 SSFT 表示为：

$$H_d(\mathbf{f}) = \begin{cases} d(\Lambda_2)/d(\Lambda_1), & \mathbf{f} \in \mathcal{V}(\Lambda_2^*) \\ 0, & \mathbf{f} \in \mathcal{V}_1^* \setminus \mathcal{V}_2^* \end{cases} \quad (4.1.9)$$

例 4.2 考虑图 4.3 中给出的下转换的例子。相应的生成矩阵是：

$$[\mathbf{V}_1]=\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix};[\mathbf{V}_2]=\begin{bmatrix}2 & 1\\0 & 2\end{bmatrix} \tag{4.1.10}$$

$$[\mathbf{U}_1]=\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix};[\mathbf{U}_2]=\begin{bmatrix}1/2 & 0\\-1/4 & 1/2\end{bmatrix} \tag{4.1.11}$$

理想的预滤波器是

$$H_d(\mathbf{f})=\begin{cases}1/4, & \mathbf{f}\in\mathcal{V}_2^*\\0, & \mathbf{f}\in\mathcal{V}_1^*\setminus\mathcal{V}_2^*\end{cases} \tag{4.1.12}$$

在图 4.3 中示于点阵 Λ_1^*。没有预滤波，原始的圆形频谱将会在 Λ_2^* 中引起混叠。通过在 Λ_1^* 中使用预滤波，只保留了包含在$\mathcal{V}_2^*$ 中的部分频谱，因而不会发生混叠。注意，如果在 $\psi_{s,1}$ 中存在混叠，那么它将保留在$\psi_{s,2}$ 中。另一方面，如果原始信号在$\mathcal{V}_2^*$ 内有一个支撑区，那么当在 Λ_1 中获取原始采样信号或对$\psi_{s,1}$ 下采样获取$\psi_{s,2}$ 时，就不需要使用预滤波。

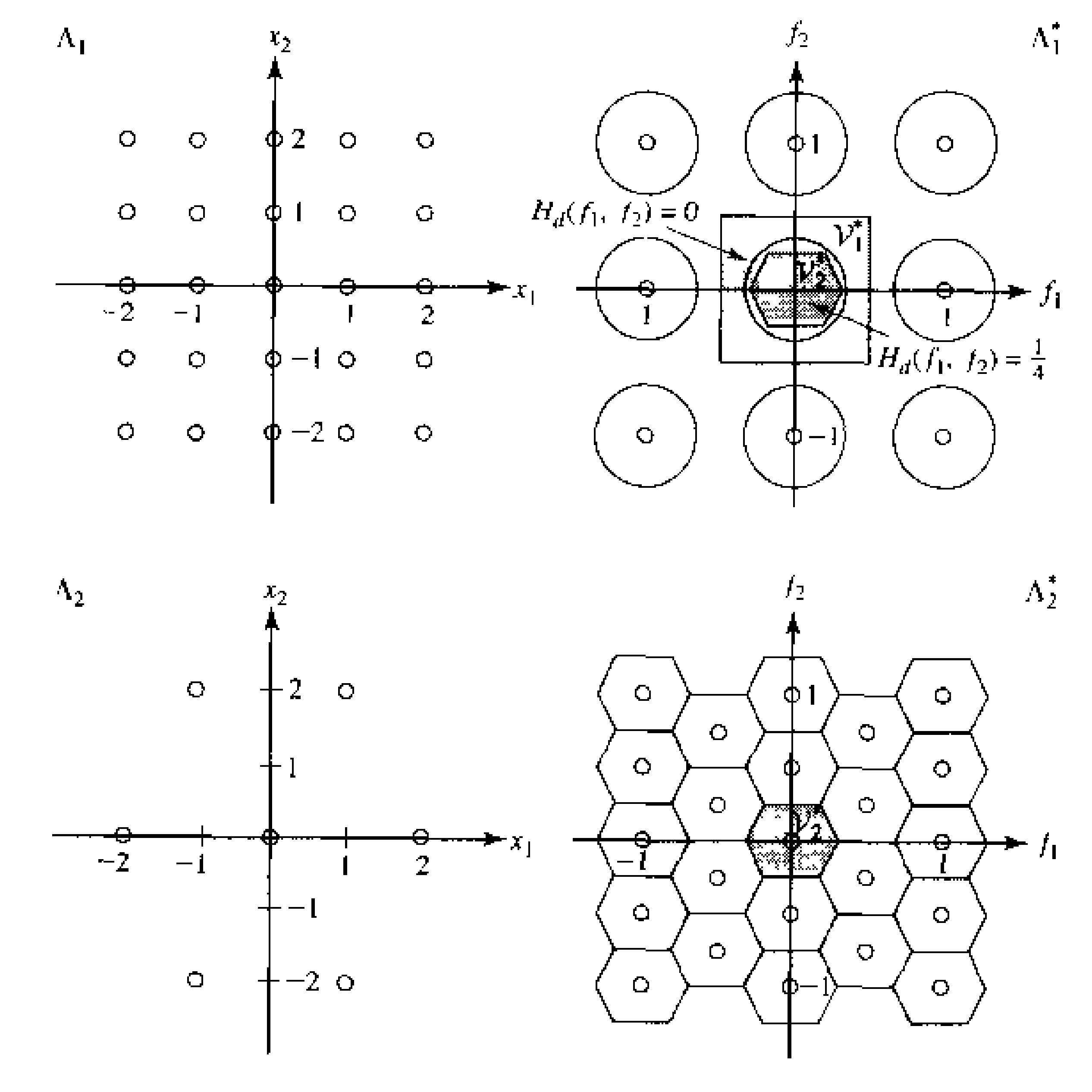


图 4.3 一个从点阵 Λ_1 到 Λ_2 下转换的例子。理想的预滤波器示于$\mathcal{V}_1^*$ 中（见例 4.2）

4.1.3 任意点阵间的转换

正如本节开始时所述，通常情况下，当 Λ_1 和 Λ_2 相互不为子集时，我们需要引入另一个点阵 Λ_3，使得 Λ_1 和 Λ_2 都是 Λ_3 的子集。显然，为使 Λ_3 的采样密度最小，我们希望找到满足上述准则的最小点阵。一旦 Λ_3 被确定，那么可以通过将前两小节讨论的上转换和下转换步骤级联，以实现从 Λ_1 到 Λ_2 的转换。在这两步中，内插滤波器和预滤波器分别为：

$$H_u(\mathbf{f})=\begin{cases}d(\Lambda_3)/\mathrm{d}(\Lambda_1), & \mathbf{f}\in\mathcal{V}_1^* \\ 0, & \mathbf{f}\in\mathcal{V}_3^*\setminus\mathcal{V}_1^*\end{cases} \tag{4.1.13}$$

$$H_d(\mathbf{f})=\begin{cases}d(\Lambda_2)/\mathrm{d}(\Lambda_3), & \mathbf{f}\in\mathcal{V}_2^* \\ 0, & \mathbf{f}\in\mathcal{V}_3^*\setminus\mathcal{V}_2^*\end{cases} \tag{4.1.14}$$

显然,这两个都定义在 Λ_3 中的滤波器可以合并为一个滤波器,具有用 SSFT 表示的如下频率响应:

$$\mathrm{H}(\mathbf{f})=\begin{cases}d(\Lambda_2)/d(\Lambda_1), & \mathbf{f}\in(\mathcal{V}_1^*\cap\mathcal{V}_2^*), \\ 0, & \mathbf{f}\in\mathcal{V}_3^*\setminus(\mathcal{V}_1^*\cap\mathcal{V}_2^*)\end{cases} \tag{4.1.15}$$

上述讨论假设可以用某种方法确定 Λ_3。根据定理 3.2,可以用 Λ_1 和 Λ_2 的和来确定 Λ_3,即 $\Lambda_3=\Lambda_1+\Lambda_2$(当$[\mathbf{V}_1]^{-1}[\mathbf{V}_2]$只包含有理数时)。回顾在一维的情况下,为能够在两个采样率之间转换,一个必要条件是,两个采样频率 $f_{s,1}$ 和 $f_{s,2}$(或者等效地,两个采样间隔 Δ_1 和 Δ_2)之间的比值必须是一个有理数;即 $r=f_{s,2}/f_{s,1}=\Delta_1/\Delta_2=p/q$,其中 p 和 q 都是整数。在这种情况下,我们可以先进行 p 倍的上采样,再进行 q 倍的下采样,从而实现 r 倍的采样率转换。显然,这只是这里所描述的更一般的处理步骤的一个特例。特别地,上述两步分别对应于从 Λ_1 到 Λ_3 的上转换和从 Λ_3 到 Λ_2 的下转换。当不满足定理 3.2 的条件时,我们可以首先确定 Λ_1 和 Λ_2 的并集,它可能不是一个合法的点阵。然后需要通过加入附加点,将此集合扩展成一个点阵,这通常是通过观察进行的。

例 4.3　考虑在两个具有如下生成矩阵的采样点阵 Λ_1 和 Λ_2 之间的转换:

$$[\mathbf{V}_1]=\begin{bmatrix}2 & 1\\ 0 & 1\end{bmatrix},\qquad [\mathbf{V}_2]=\begin{bmatrix}2 & 1\\ 0 & 2\end{bmatrix} \tag{4.1.16}$$

它们对应的反商点阵的生成矩阵是:

$$[\mathbf{U}_1]=\begin{bmatrix}1/2 & 0\\ -1/2 & 1\end{bmatrix},\qquad [\mathbf{U}_2]=\begin{bmatrix}1/2 & 0\\ -1/4 & 1/2\end{bmatrix} \tag{4.1.17}$$

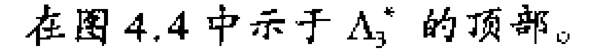
首先,我们根据上述生成矩阵绘出输入和输出采样点阵 Λ_i 以及反商点阵 Λ_i^*, $i=1,2$,如图 4.4所示。在这种情况下,Λ_1 和 Λ_2 并非彼此的子集。因此,我们需要确定一个点阵 Λ_3,它是包含 Λ_1 和 Λ_2 的最小的点阵。因为$[\mathbf{V}_1]^{-1}[\mathbf{V}_2]$只含有理数,所以 $\Lambda_3=\Lambda_1+\Lambda_2$。一般,通过分析确定 Λ_3 的生成矩阵是困难的。然而,通过观察,我们可以看到在这种情况中 Λ_3 一定是图 4.4 中画出的方形点阵。对于这个点阵,一个明显的生成矩阵是单位矩阵,即:

$$[\mathbf{V}_3]=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \tag{4.1.18}$$

反商点阵也是方形的,其生成矩阵也是单位矩阵,即$[\mathbf{U}_3]=([\mathbf{V}_3]^T)^{-1}=[\mathbf{V}_3]$。

由画出的反商点阵 Λ_i^*, $i=1,2,3$,我们可以确定它们各自的沃罗纳晶格,如图 4.4 所示。在这种情况中,$\mathcal{V}_1^*\supset\mathcal{V}_2^*$,从而$\mathcal{V}_1^*\cap\mathcal{V}_2^*=\mathcal{V}_2^*$。因此,中间滤波器应该是:

$$H(f)=\begin{cases}\dfrac{d(\Lambda_2)}{d(\Lambda_1)}=\dfrac{1}{2}, & \mathbf{f}\in\mathcal{V}_2^* \\ 0, & \mathbf{f}\in\mathcal{V}_3^*\setminus\mathcal{V}_2^*\end{cases}$$



在图 4.4 中示于 Λ_3^* 的顶部。

图 4.4 两个互不包含的点阵 Λ_1 和 Λ_2 之间的转换。Λ_3 是中间的点阵，包含了 Λ_1 和 Λ_2。理想的滤波器示于 Λ_3^* 中(见例 4.3)

4.1.4 滤波器的设计与实现以及其他内插方法

在前面的讨论中，我们只是用 SSFT 规定所期望的滤波器。在空间域的等效滤波器 $h(\mathbf{x})$，$\mathbf{x}\in\Lambda_3$ 可以由其逆 SSFT 确定。从理论上来说，为实现滤波运算，我们首先需要将输入信号从 Λ_1 向上采样到 Λ_3，在 Λ_3 中进行滤波，然后再将其从 Λ_3 向下采样到 Λ_2，如图 4.1(c)所示。在 Λ_3 中的滤波运算是：

$$\Psi_{s,4}(\mathbf{x}) = \psi_{s,3}(\mathbf{x}) * h(\mathbf{x}) = \sum_{\mathbf{y}\in\Lambda_3} \mathcal{U}\{\psi_{s,1}\}(\mathbf{y})h(\mathbf{x}-\mathbf{y}), \quad \mathbf{x}\in\Lambda_3 \quad (4.1.19)$$

实际上，不需要对 Λ_3 中所有的样点计算滤波输出，因为只有在 Λ_2 中的样点才是所期望的输出。另外，由于上采样的输入信号只在 Λ_1 中是非零的，所以公式(4.1.19)中的和仅包含 Λ_1 中的输入样点。总之，Λ_2 上的输出信号可以由下式直接从 Λ_1 上的输入获得：

$$\psi_{s,2}(\mathbf{x}) = \mathcal{D}\{\psi_{s,4}\}(\mathbf{x}) = \sum_{\mathbf{y}\in\Lambda_1} \psi_{s,1}(\mathbf{y})h(\mathbf{x}-\mathbf{y}), \quad \mathbf{x}\in\Lambda_2 \quad (4.1.20)$$

这意味着，样点 $\mathbf{x}\in\Lambda_2$ 是用样点 $\mathbf{y}\in\Lambda_1$ 的加权平均估计的，权重等于 $h(\mathbf{x}-\mathbf{y})$。

在前面的讨论中，我们简单地给出了预滤波器和重建滤波器应有的频率响应，并没有考虑如何设计这些滤波器。根据公式(4.1.15)所指定的频率响应 $H(\mathbf{f}),\mathbf{f}\in\mathcal{V}_3^*$，我们可以通过逆SSFT来获得理想的滤波器 $h(\mathbf{x}),\mathbf{x}\in\Lambda_3$。然而，由于 $H(\mathbf{f})$具有很陡峭的过渡带，所以这个滤波器将具有无限长冲激响应。如果我们简单地截断滤波器，将会在滤波后的信号中产生讨厌的振铃现象。因此，必须用一个合适的窗函数来仔细地修改所需的滤波响应，以便使相应的滤波器具有有限长度冲激响应。我们也可以用各种优化技术来直接设计有限长脉冲响应滤波器，使它们的频率响应尽可能接近期望的响应。滤波器设计技术超出了本书的范围，有兴趣的读者可以参阅参考文献[4]，其中对多维滤波器的设计有很好的介绍。绝大多数数字滤波器设计技术是基于用离散频率表示频率响应的假设。这里，为了解决这个问题，必须把用连续频率表示的所需频率响应转换为用离散频率表示，如公式(3.2.16)所规定。根据转换后的频率响应，我们可以采用数字滤波器设计技术得到所期望的离散空间的滤波器。我们可以在抽象的离散空间或直接在采样空间实现卷积运算。

在下文中将会看到，对于视频信号的采样和采样率转换，为降低计算的复杂性，经常使用的是与理想频率响应相距甚远的简单滤波器。在 4.2 节描述了一些用于重要转换问题的滤波器示例。

到目前为止，我们已经基于频域考虑导出了所需的内插滤波器。我们也可以基于采样域需求提出内插方法。对于由采样点进行连续信号内插，问题在于通过给定的采样值构造一个连续表面。对于上转换来说，问题在于由周围的已知样点估计出丢失的样点值。一般方法是在每一个小的区域用一个有限阶多项式函数来逼近这个连续信号，并且根据该区域给定的采样值用最小均方匹配处理步骤来确定多项式系数。一旦这个多项式被确定，我们既可以重建整个连续表面，也可以得到丢失的采样点。为了保持在相邻区域所得到的表面的连续性，可以使用样条函数，它是一类特殊的分段多项式(最流行的是为实现二维信号内插所描述的三次样条方法[6])。

4.2　视频信号的采样率转换

在视频系统中经常需要采样率转换。例如，在一个 NTSC 系统中显示 PAL 信号，需要将一个采样率为 $f_{s,t}=50$ 场/秒，$f_{s,y}=625$ 行/图像高度的隔行信号转换成一个 $f_{s,t}=60$ 场/秒，$f_{s,y}=525$ 行/图像高度的隔行信号。为了将胶片上的电影转换为 NTSC TV 广播和显示，需要从 $f_{s,t}=24$ 帧/秒(fps)到 $f_{s,t}=60$ 场/秒的转换。另一个有趣的问题是将隔行扫描转换成逐行扫描，称为去隔行，它是在采用逐行扫描的计算机屏幕上显示专为电视制作的节目时所需要的。一些

先进的电视系统已经具有了内置的去隔行功能，使得隔行的节目能够以逐行的方式显示，从而提高视觉质量。

如4.1节中所述，解决采样率转换问题的一般步骤如下：(1)确定输入和输出信号的等效采样点阵，以及一个覆盖输入和输出信号所有采样点的中间点阵；(2)基于这三个点阵的沃罗纳晶格，确定所需的滤波器频率响应；(3)设计一个近似于所需响应的滤波器。实际上，由于视频信号具有很高的码率，我们不能使用具有很多系数的滤波器，特别是在时间方向上。为了降低计算复杂度，我们也可以将空时转换问题分解成空间转换后跟时间转换，或者反之。

下面，我们更详细地描述去隔行以及在PAL与NTSC之间转换的解决方法。我们只考虑在垂直和时间方向上的运算，因为水平方向的采样率转换可以用传统的一维技术分别完成。Haskell，Puri和Netravali [5]对各种视频转换问题和它们的实际解决方法进行了广泛的讨论，这些问题中的一部分在本章末尾的习题部分予以考虑。

4.2.1 去隔行

去隔行的问题是在每一场中填补被跳过的那些行，如图4.5所示。考虑隔行的NTSC信号，它的场率为 $f_{s,t}=60$ 场/秒，行率为 $f_{s,y}=525$ 行/图像高度。采样间隔是 $\Delta_t=1/60$ 秒，$\Delta y=1/525$图像高度，相应的采样点阵和它的反商如图4.6(a)所示。去隔行信号的采样点阵是一个方形的点阵，如图4.6(b)所示。与前面一样，我们对垂直和时间轴进行了定标，以使 Δ_y 和 Δ_t 用同样的长度表示。类似地，$f_{s,y}$和$f_{s,t}$在频域也被赋予相同的长度。

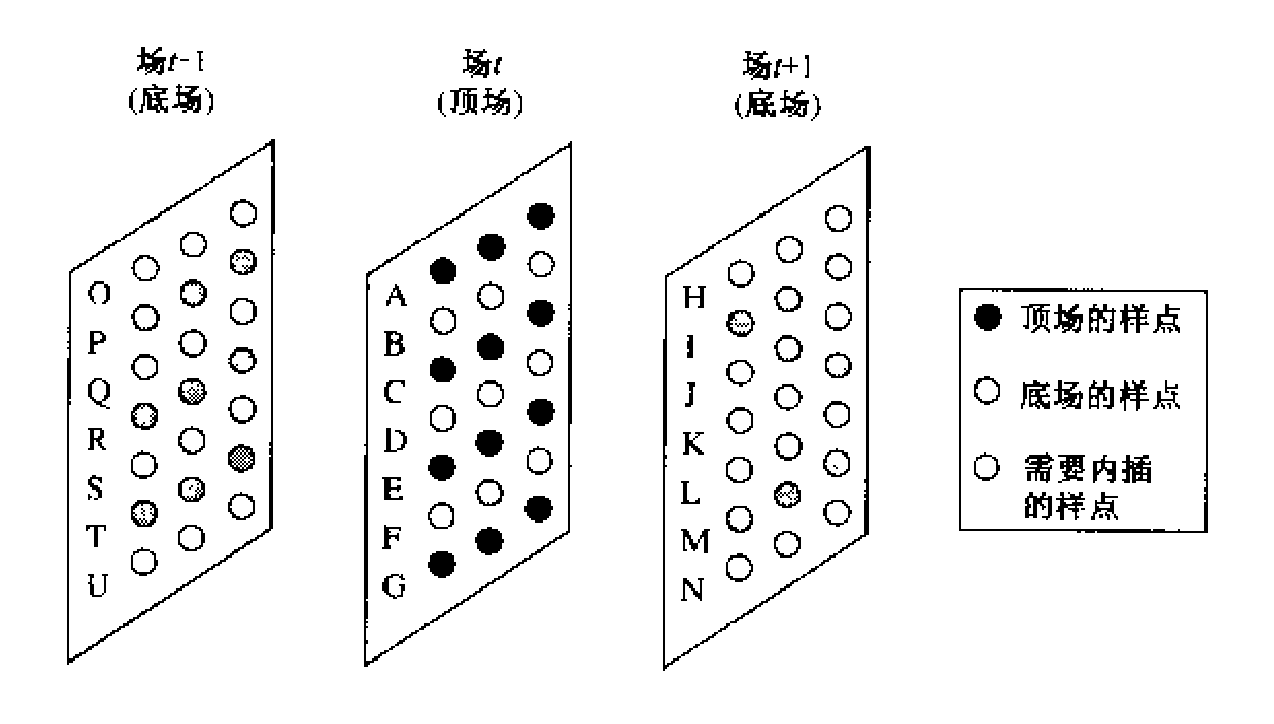


图4.5 去隔行过程；场 t 和场 $t+1$ 形成一个隔行的帧

显然，这两个点阵及其反商点阵的生成矩阵是：

$$[\mathbf{V}_1]=\begin{bmatrix}2\Delta_t & \Delta_t\\ 0 & \Delta_y\end{bmatrix},\qquad [\mathbf{U}_1]=\begin{bmatrix}1/2\Delta_t & 0\\ -1/2\Delta_y & 1/\Delta_y\end{bmatrix}\tag{4.2.1}$$

$$[\mathbf{V}_2]=\begin{bmatrix}\Delta_t & 0\\ 0 & \Delta_y\end{bmatrix},\qquad [\mathbf{U}_2]=\begin{bmatrix}1/\Delta_t & 0\\ 0 & 1/\Delta_y\end{bmatrix}\tag{4.2.2}$$

在这种情况中，$\Lambda_2\supset\Lambda_1$，这个问题是一个上转换问题，理想的内插滤波器是：

$$H(f_y,f_t)=\begin{cases}\dfrac{d(\Lambda_2)}{d(\Lambda_1)}=2, & (f_y,f_t)\in\mathcal{V}_1^*\\ 0, & (f_y,f_t)\in\mathcal{V}_2^*\setminus\mathcal{V}_1^*\end{cases}\tag{4.2.3}$$

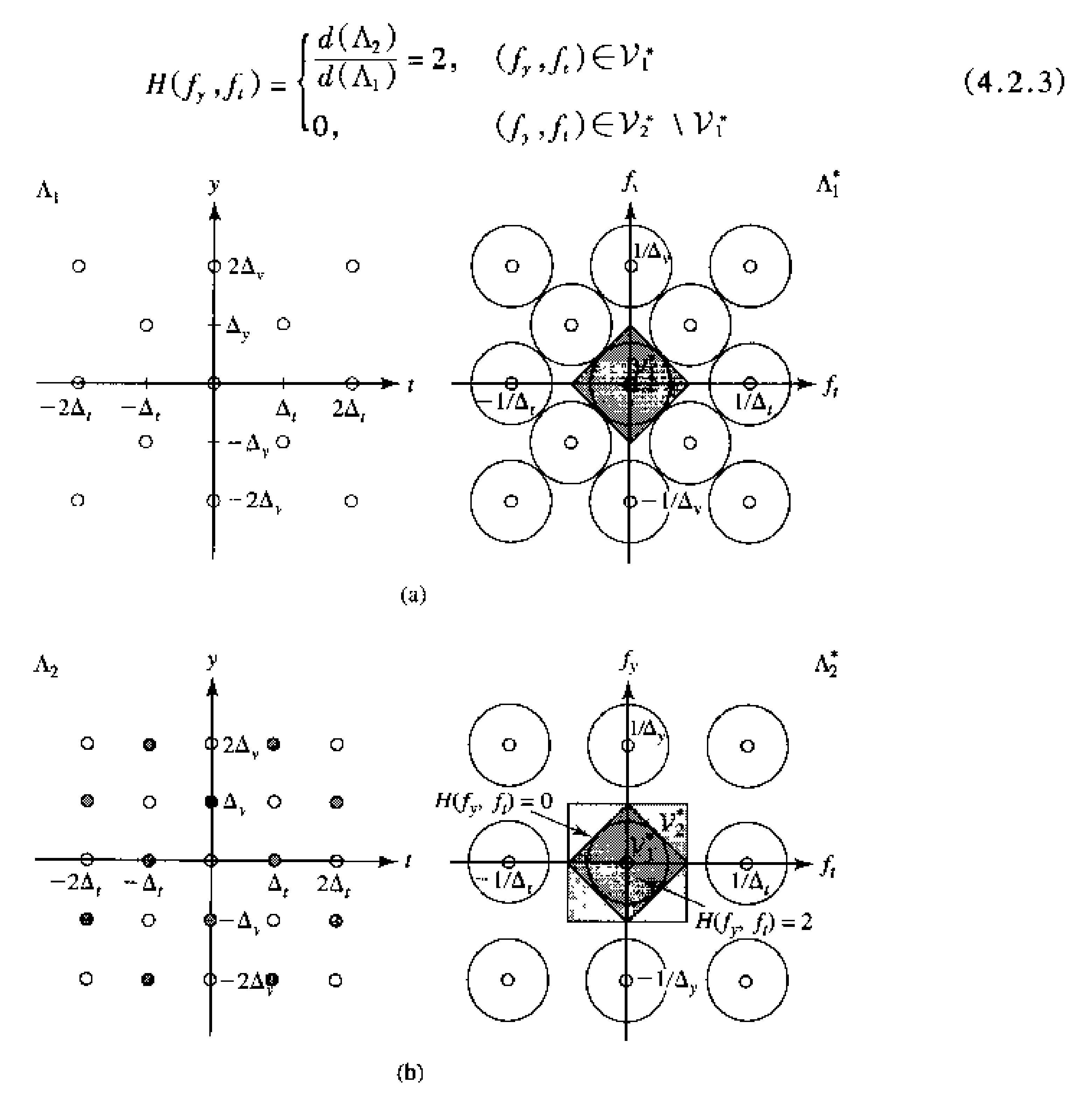


图4.6　去隔行前后的采样点阵和它们的反商点阵。空心圆圈代表原始样点，实心圆圈代表要内插的样点。(a)相应于隔行扫描的采样点阵及其反商点阵；(b)去隔行后的采样点阵及其反商点阵

在图4.6中也示出了沃罗纳晶格$\mathcal{V}_1^*$和$\mathcal{V}_2^*$。此滤波器频率响应的幅度示于图4.7(a)。我们可以看到这是一个具有菱形通带的低通滤波器。注意，这个滤波器是不可分离的(也就是说，不能被分解成时间滤波器和垂直滤波器的乘积)，因此必须使用二维滤波器设计技术。为了简化设计问题，我们可以首先设计一个具有方形通带的可分离的低通滤波器，然后对所得的滤波器进行旋转。

实际上，为实现去隔行已经提出了很多较简单的滤波器。一种选择是用同一场中的垂直内插，这是个一维二倍上转换问题。理想的垂直滤波器是一个半带低通滤波器[7]；然而，这个滤波器要求无限长度冲激响应，而且是不可实现的。在实际中用的是短得多的滤波器。最简单的一种是行平均，它是用丢失行的上一行和下一行的平均来估计该丢失行。例如，在图4.5中，对于第t场，$D=(C+E)/2$。Λ_2中的等效滤波器是：

$$h(y,t)=\begin{cases}1, & (y,t)=(0,0),\\ 1/2, & (y,t)=(\Delta_y,0),(-\Delta_y,0),\\ 0, & 其他\end{cases}\tag{4.2.4}$$

应用公式(3.2.3),此滤波器的频率响应(即 Λ_2 上 $h(y,t)$ 的 SSFT)是:

$$H(f_y,f_t)=1+\frac{1}{2}(e^{j2\pi\Delta_y f_y}+e^{-j2\pi\Delta_y f_y})=1+\cos(2\pi\Delta_y f_y) \tag{4.2.5}$$

这个频率响应如图 4.7(b)所示。我们可以看到,它与图(a)所示的所需响应有显著的差别。由于没有使用时域滤波,所以它沿时间频率轴具有全通特性。沿垂直频率轴,它与理想半带低通滤波器相差很远。

为了改进性能,可以使用更长的垂直内插滤波器,其频率响应更为接近理想的半带低通滤波器。例如,在图 4.5 中,对于第 t 场的行,满意的内插方法是 $D=(A+7C+7E+G)/16$。在去隔行格栅(Λ_2 中的垂直轴)上等效的内插滤波器为:

$$h=[1,0,7,16,7,0,1]/16 \tag{4.2.6}$$

按照与前面相同的方法,很容易证明这个滤波器的频率响应是:

$$H(f_y,f_t)=1+\frac{7}{8}\cos(2\pi\Delta_y f_y)-\frac{1}{8}\cos(6\pi\Delta_y f_y) \tag{4.2.7}$$

如图 4.7(c)所示。

前面的两种方法只使用了垂直内插。一种替代方法是使用时间内插。注意,对于一场中每一丢失的行,在同一帧的另一场中有一个对应行(见图 4.5)。一个简单的时间内插方案是复制此对应行。即 $D=K,J=C$。这种方法称为场合并,因为每一个去隔行帧都由合并两场获得。用于顶场去隔行的等效滤波器(假设它在底场之前出现)是:

$$h(y,t)=\begin{cases}1, & (y,t)=(0,0),(0,-\Delta_t)\\0, & \text{其他}\end{cases} \tag{4.2.8}$$

用于底场的滤波器是上述滤波器的时间反转形式。公式(4.2.8)中滤波器的频率响应是:

$$H(f_y,f_t)=1+e^{-j2\pi\Delta_t f_t} \tag{4.2.9}$$

由于这个滤波器在时间上是不对称的,所以它具有复频率响应。另外,在一帧中两场的时间内插方向是相反的(对于某些特殊图案可能会产生视觉人为失真)。其幅度响应示于图 4.7(d),同样,它与所需的响应有很大差别。由于只在时间方向上进行了滤波,因此在垂直方向上是全通的。

为了改进性能,可以使用一种对称的滤波器——例如,对前一场和后一场中的对应行取平均来获得当前场中丢失的行。例如,对于场 t,$D=(K+R)/2$。我们称这种方法为场平均。等效的内插滤波器为:

$$h(y,t)=\begin{cases}1, & (y,t)=(0,0)\\1/2, & (y,t)=(0,\Delta_t),(0,-\Delta_t)\\0, & \text{其他}\end{cases} \tag{4.2.10}$$

或者

$$H(f_y,f_t)=1+\cos(2\pi\Delta_t f_t) \tag{4.2.11}$$

其频率响应示于图 4.7(e)。尽管这个滤波器与前一个滤波器相比具有较好的频率响应,然而内插任何一场需要涉及到三个场。这需要两帧存储器,与场合并方法相比,它在存储器和延时上要求有不容忽视的增加。

为了在时间和空间人为失真方面达到折中,较好的方法是既用垂直内插也用时间内插。例如,通过对同一场中上一个和下一个像素以及前一场和后一场中对应的像素取平均,我们可

以内插出一个丢失的像素。在图 4.5 中，这相当于 $D=(C+E+K+R)/4$。这种方法被称为行和场平均。在去隔行点阵上的等效滤波器为：

$$h(y,t)=\begin{cases}1, & (y,t)=(0,0)\\ 1/4, & (y,t)=(\Delta_y,0),(-\Delta_y,0),(0,\Delta_t),(0,-\Delta_t)\\ 0, & \text{其他}\end{cases} \tag{4.2.12}$$

其频率响应为：

$$H(f_y,f_t)=1+\frac{1}{2}(\cos(2\pi f_y\Delta_y)+\cos(2\pi f_t\Delta_t)) \tag{4.2.13}$$

如图 4.7(f)所示。我们可以看到它比所有前面的滤波器更接近所需的响应。由于使用场平均，这种方法需要存储两帧。在习题 4.1 中讨论了只需存储一帧的方法。

当成像的景物在相邻两场之间静止时，在奇数场中丢失的偶数行应该与前一个和后一个偶数场中对应的偶数行完全一样。因此时间内插将产生精确的估计。另一方面，当景物中存在运动时，相邻场中对应行可能不对应同一个物体位置，时间内插将会产生不可接受的人为失真。想像一个水平移动的垂直格栅图案，在两个相邻场摄取的格栅图案移动半个格栅间隔。场平均将产生锯齿型的垂直格栅。而同时使用空间和时间平均的方法将产生不太严重的人为失真，但是所得到的视频对于某些应用来说可能仍是不可接受的。为了得到更好的质量，应该使用运动补偿内插，这将在 4.2.3 节中讨论。

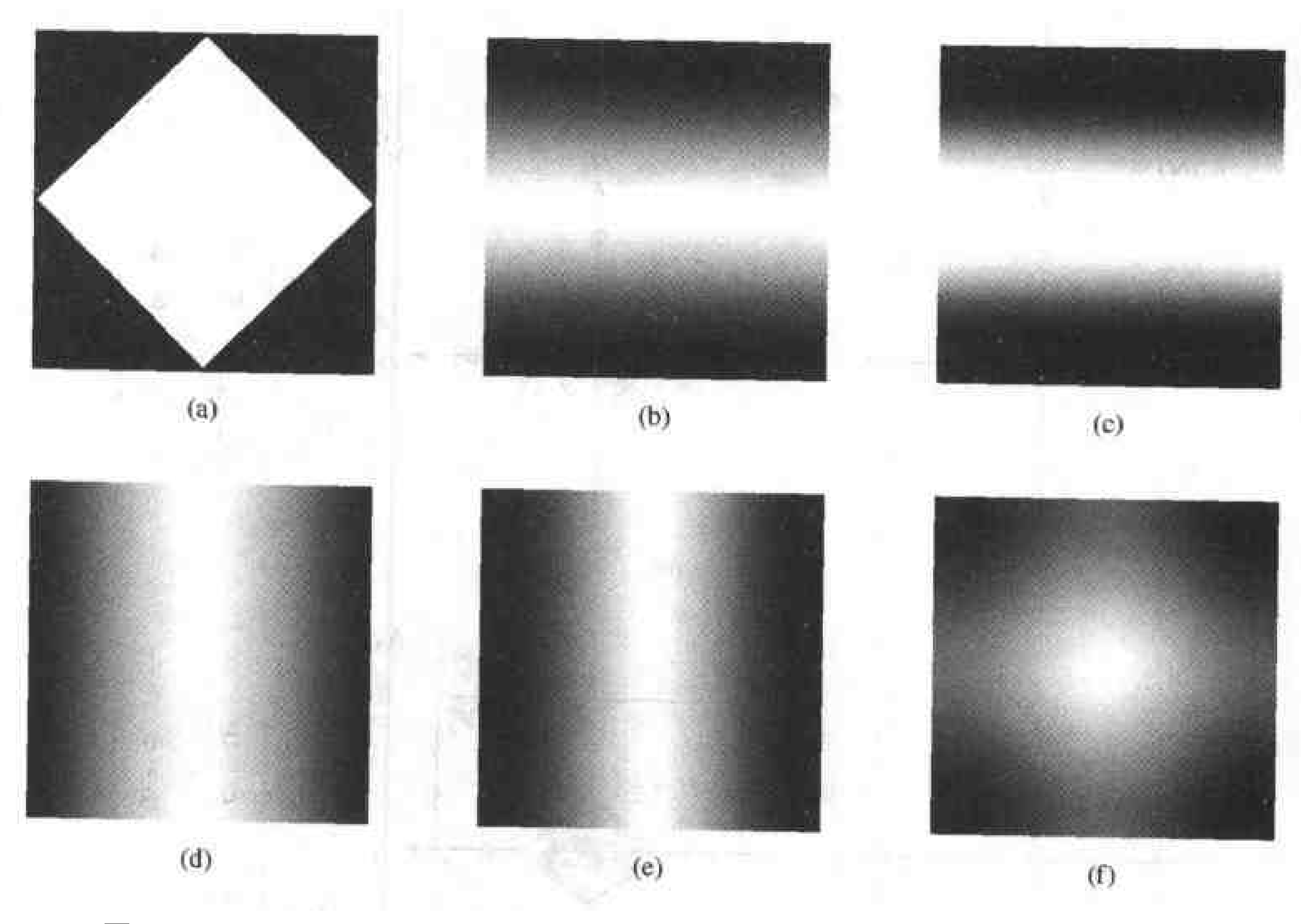


图 4.7 几种去隔行滤波器的频率响应：(a)理想的滤波器(公式 4.2.3)；(b)行平均(公式 4.2.5)；(c)四点垂直内插(公式 4.2.7)；(d) 场合并(公式 4.2.9)；(e) 场平均(公式 4.2.11)；(f) 行-场平均(公式 4.2.13)。在每一幅图中，垂直轴代表垂直频率，水平轴代表时间频率。频率范围是 $|f_y|\leq f_{s,y}/2$，$|f_t|\leq f_{s,t}/2$

4.2.2 PAL与NTSC信号之间的转换

在这一节中，我们考虑从PAL到NTSC信号制式的转换（都是隔行扫描）。从NTSC到PAL的转换也遵循同样的原理，在习题4.4中考虑了这个问题。回顾PAL信号的行率和场率为$f_{y,1}=625$行/图像高度，$f_{t,1}=50$场/秒；而NTSC信号是$f_{y,2}=525$行/图像高度，$f_{t,2}=60$场/秒。PAL(Λ_1)和NTSC(Λ_2)制式的采样点阵分别画在图4.8(a)和(b)中，其中$\Delta_t=1/60$秒，$\Delta_y=1/525$图像高度。根据每个采样点阵上的点，我们可以很容易地确定它们的生成矩阵，从而确定[U]和[V]矩阵。根据[U]矩阵，可以确定它的反商点阵和它的沃罗纳晶格。所有这些都在图4.8中给出。在绘制点阵时，我们已经对时间和垂直轴进行了定标，方法是使空间-时间域的Δ_t和Δ_y相等，使频域的$1/\Delta_t$和$1/\Delta_y$相等。

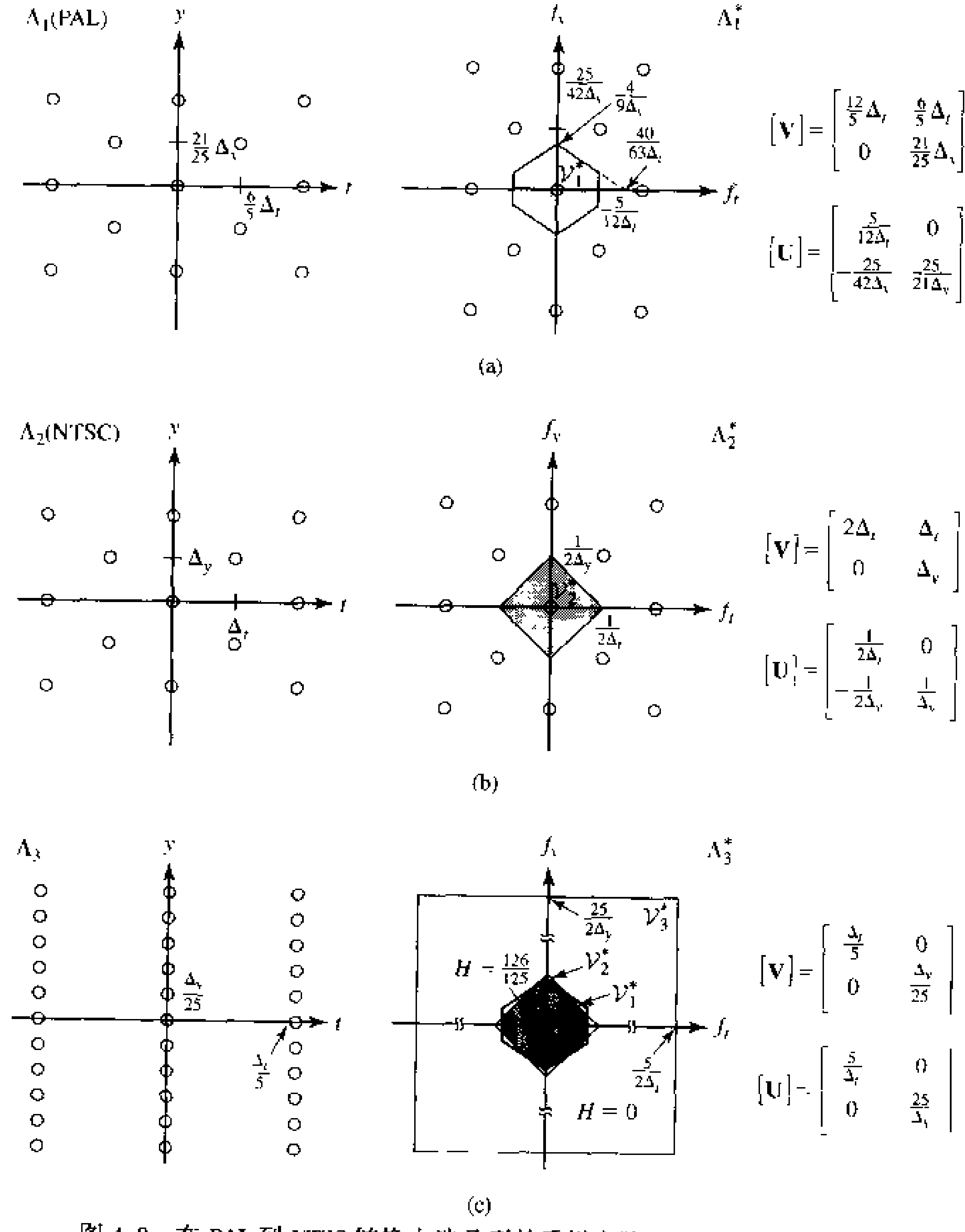


图4.8 在PAL到NTSC转换中涉及到的采样点阵。Λ_1和Λ_2是PAL和NTSC制式的采样点阵，Λ_3是既包含Λ_1也包含Λ_2的中间点阵

PAL 与 NTSC 之间的转换是困难的，因为这两个点阵的垂直和时间采样率不是互为整数倍。首先，我们用 4.1 节中描述的技术来获得理想的解决方案。为此目的，我们需要确定一个既包含 Λ_1 也包含 Λ_2 的点阵 Λ_3。通过观察，$f_{y,1}=625$ 和 $f_{y,2}=525$ 的最小公倍数是 $f_{y,3}=13\ 125$，而且 $f_{t,1}=50$ 和 $f_{t,2}=60$ 的最小公倍数为 $f_{t,3}=300$，可以看出，行率为 $f_{y,3}$ 和帧率为 $f_{t,3}$ 的矩形点阵是所需的中间点阵，即为图 4.8(c)中的点阵 Λ_3。从图中可以容易地看出，在 Λ_3 上滤波器的所需频率响应为：

$$H(f_y,f_t)=\begin{cases}\dfrac{d(\Lambda_2)}{d(\Lambda_1)}=126/125, & (f_y,f_t)\in\mathcal{V}_1^*\cap\mathcal{V}_2^*\\ 0, & (f_y,f_t)\in\mathcal{V}_3^*\setminus(\mathcal{V}_1^*\cap\mathcal{V}_2^*)\end{cases}\tag{4.2.14}$$

这个理想的转换滤波器示于图 4.8(c)的 Λ_3^* 中(在图 4.11(a)也将看到)。显然，由于 $\mathcal{V}_1^*\cap\mathcal{V}_2^*$ 的形状非常复杂，所以这个滤波器不容易设计。

前面提出的解决方案要求直接把信号从 Λ_1 转换到 Λ_3，然后返回到 Λ_2，如图 4.9(a)所示。实际上，这个问题通常是用四个步骤解决的：(1)对 PAL 信号的每一场进行去隔行，以使产生的每一帧包含 625 行，(2)在每一个去隔行的帧中将行率从 625 向下转换到 525 行，(3)将场率从50 Hz向上转换到 60 Hz，(4)最后将每一帧分裂成两个隔行的场。此顺序的实现步骤示于图 4.9(b)。

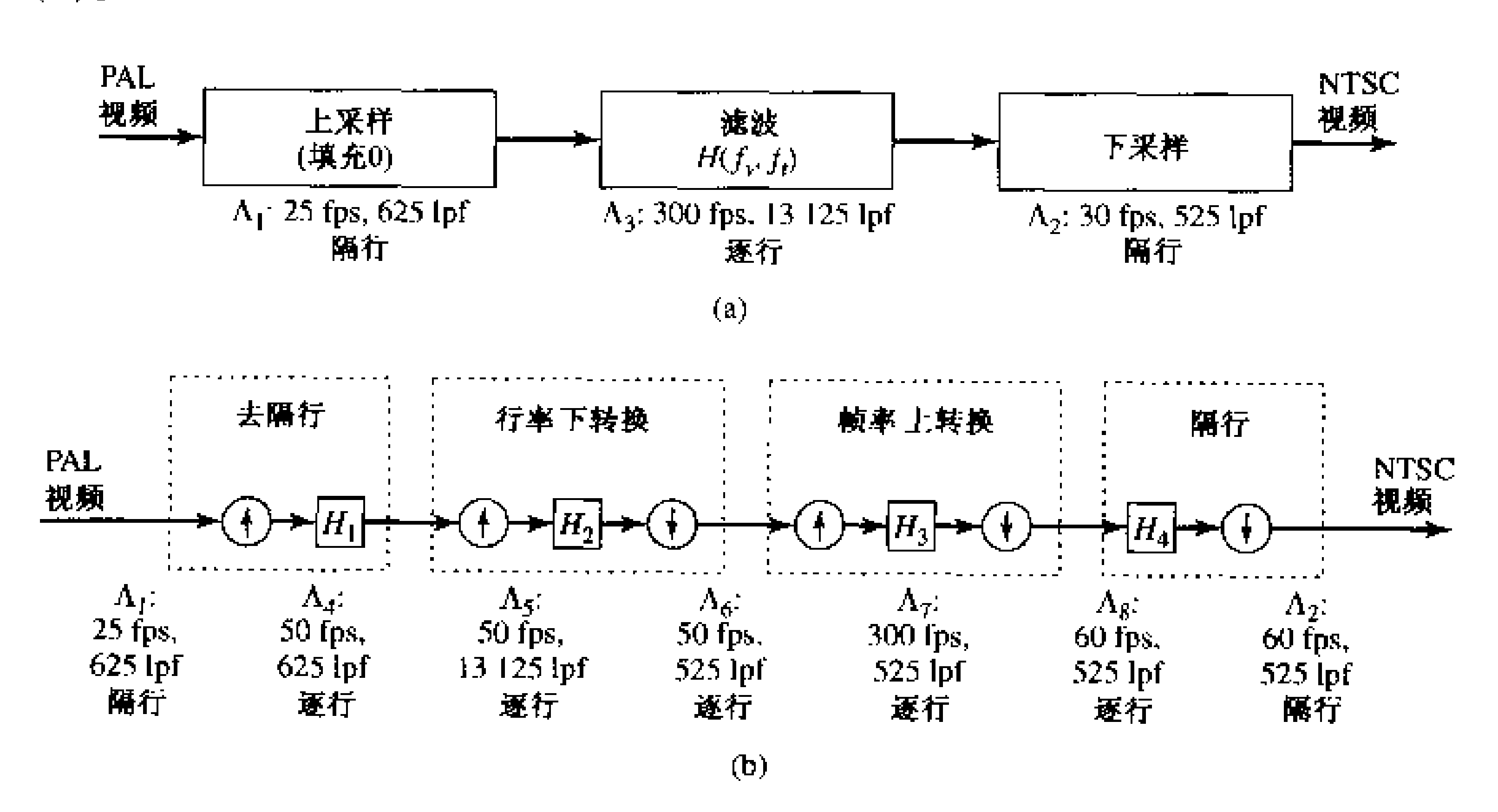
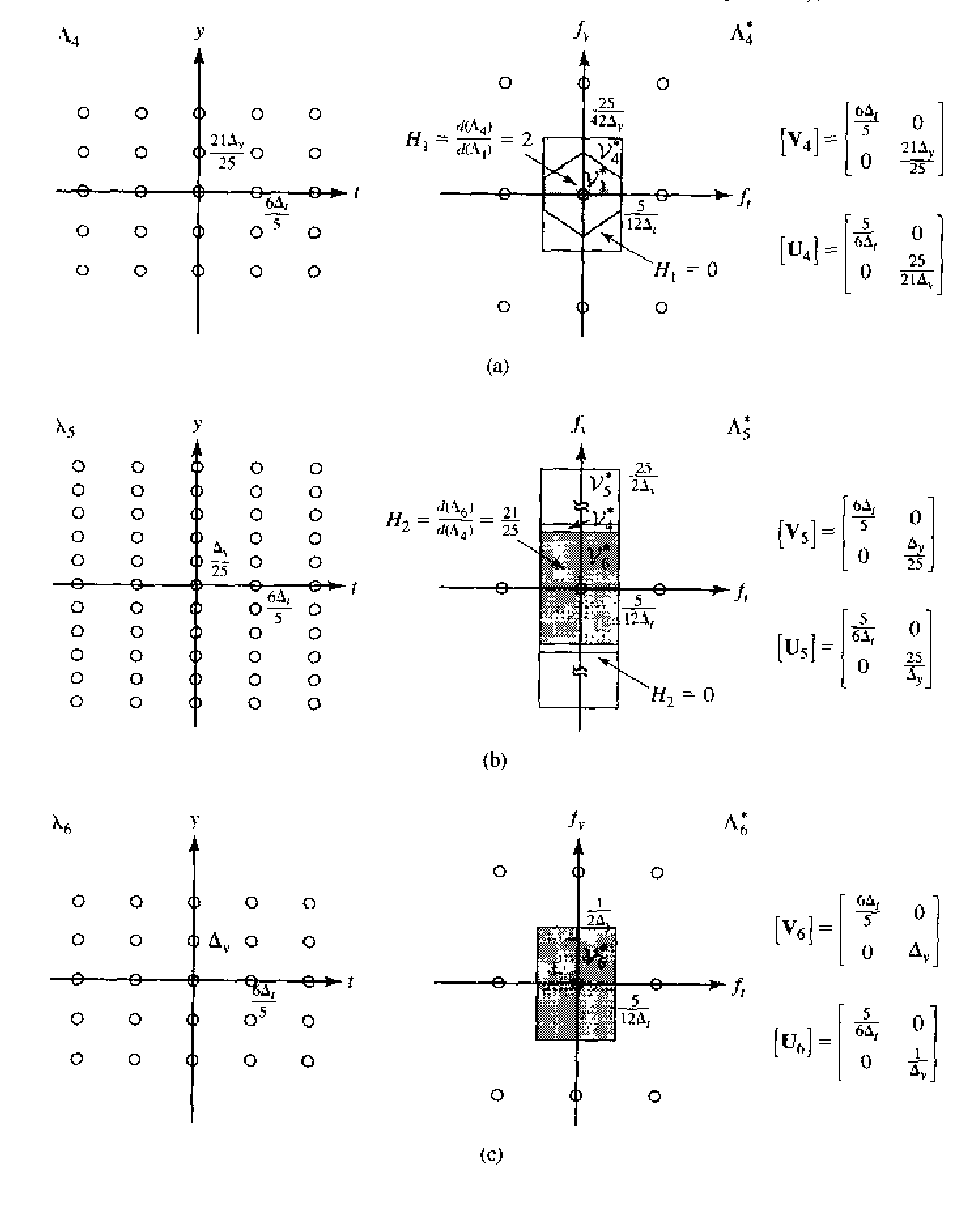


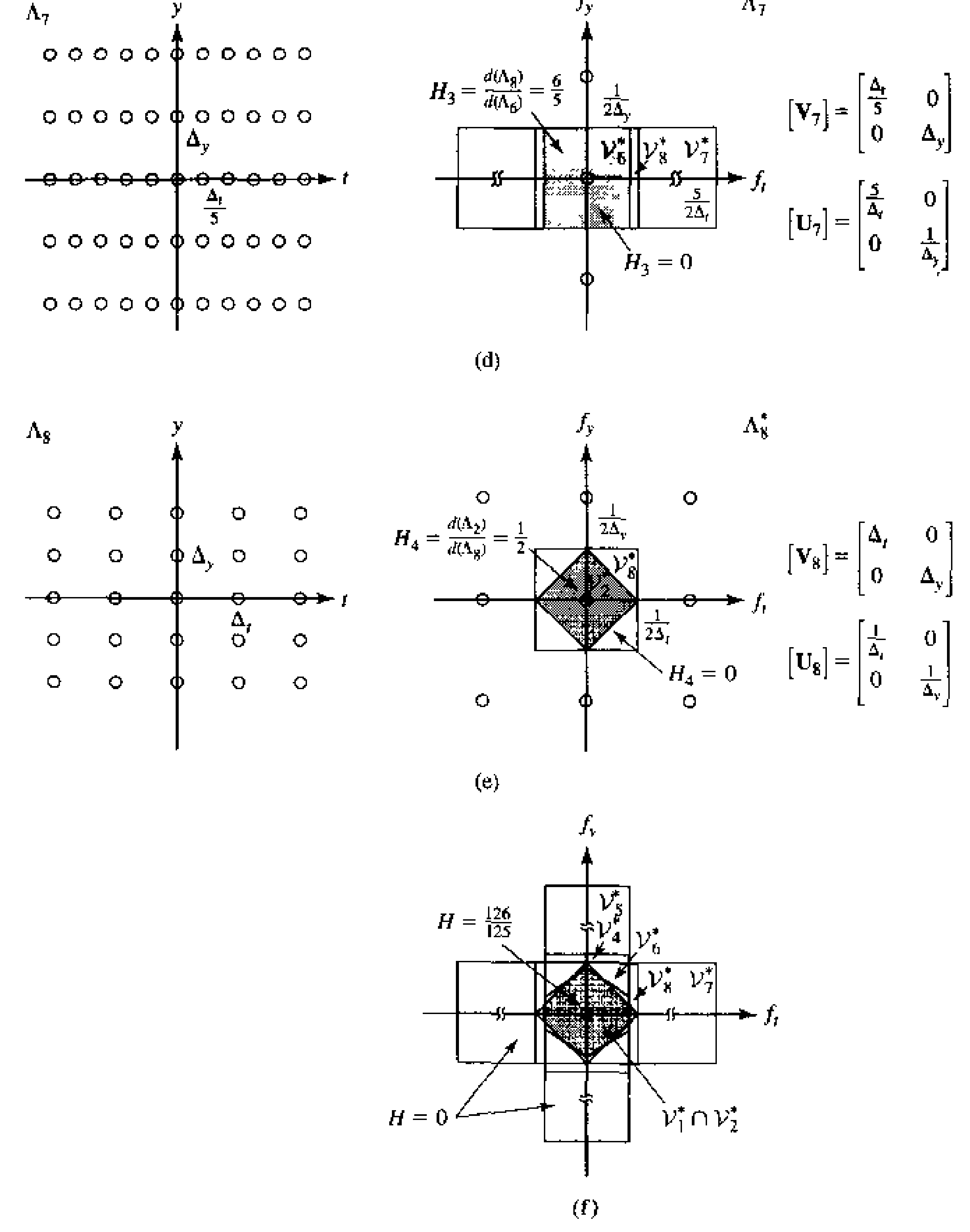
图 4.9　从 PAL 转换到 NTSC 制式的两种不同的方法：
(a)直接转换　(b)顺序实现

在 4.2.1 节中已经讨论了一般的去隔行问题，而且对指定设置的解决方案示于图 4.10(a)。对于从 625 行到 525 行的垂直转换，理想情况下我们应该首先把行率转换到 13 125 行，在此高分辨率下对信号进行滤波，然后再向下采样到 525 行，如图 4.9(b)所示。理想的解决方案示于图 4.10 (a～c)。对于从 50 到 60 fps 的时间转换，我们必须首先将帧率转换到 300 fps，在此高分辨率下对信号进行滤波，然后再向下采样到 60 fps，如图 4.9(b) 所示。理想的解决方案示于图 4.10(c～e)。最后，为了从逐行帧中生成隔行信号，我们可以简单地在每一帧中每两行跳过一行。但是为了避免混叠，需要一个预滤波器，该解决方案在图 4.10(e)中给

出。总的等效滤波器响应示于图4.10(f)。与图 4.8(a)所示的理想滤波器相比较,这个滤波器具有相同的通带,但是具有不同的阻带。对应于 H_2 和 H_3 级联的频率响应示于图 4.11(b)。尽管直接转换和顺序方法得到相同的频率响应,然而可以证明顺序操作能够降低计算的需求(见习题 4.3)。

注意,将 625 行转换到 525 行等效于将每 25 行转换为 21 行[13]。如果不使用在 13 125 行分辨率下的理想内插滤波器,一种简便的方法是对于需要确定的 21 行中的每一行,在给定的 25 行中找到两个最接近的相邻行,并且用线性内插确定采样值,如图 4.12 所示。为了确定与这两个已知行有关的内插系数,我们应该确定它们与要内插行的距离。如果与左边已知行的距离为 d_l,与右边已知行的距离为 d_r,那么系数应该分别是 $h_l = d_r/(d_l + d_r)$ 和 $h_r = d_l/(d_r + d_l)$。令 $\Delta_{y,3}$代表在 13 125 行格栅中每两行之间的间距,$\Delta_{y,1}$和 $\Delta_{y,2}$分别代表在 625 行





(d)

(e)

(f)

图 4.10　在从 PAL 到 NTSC 转换的顺序实现中涉及的理想操作

和 525 行格栅中的间距。显然，$\Delta_{y,1}=21\Delta_{y,3}$，$\Delta_{y,2}=25\Delta_{y,3}$。对于在 525 行格栅中的行 l，如果 $k=l*\Delta_{y,2}/\Delta_{y,1}=\frac{25}{21}l$ 是一个整数的话，那么就可以直接将 625 行格栅中的行 k 复制过来。当 $l=0,21,42,\ldots$ 时会发生这种情况。对于在两行中间的行来说，在 625 行格栅中左右相邻行的标号分别为 $k=\lfloor l*\Delta_{y,2}/\Delta_{y,1}\rfloor$ 和 $k+1$。相应的距离是 $d_l=l\Delta_{y,2}-k\Delta_{y,1}$ 和 $d_r=(k+1)\Delta_{y,1}-l\Delta_{y,2}$。因此，内插系数是 $h_l=k+1-l*25/21$ 和 $h_r=l*25/21-k$。图 4.12 给出了每个内插行的左边已知行的权重。

在 Λ_3 中等效的垂直滤波器是：

$$h_v(y)=\begin{cases}(1-|n|/21), & y=n\Delta_{y,3},|n|=0,1,\cdots,20\\0 & \text{其他}\end{cases} \tag{4.2.15}$$

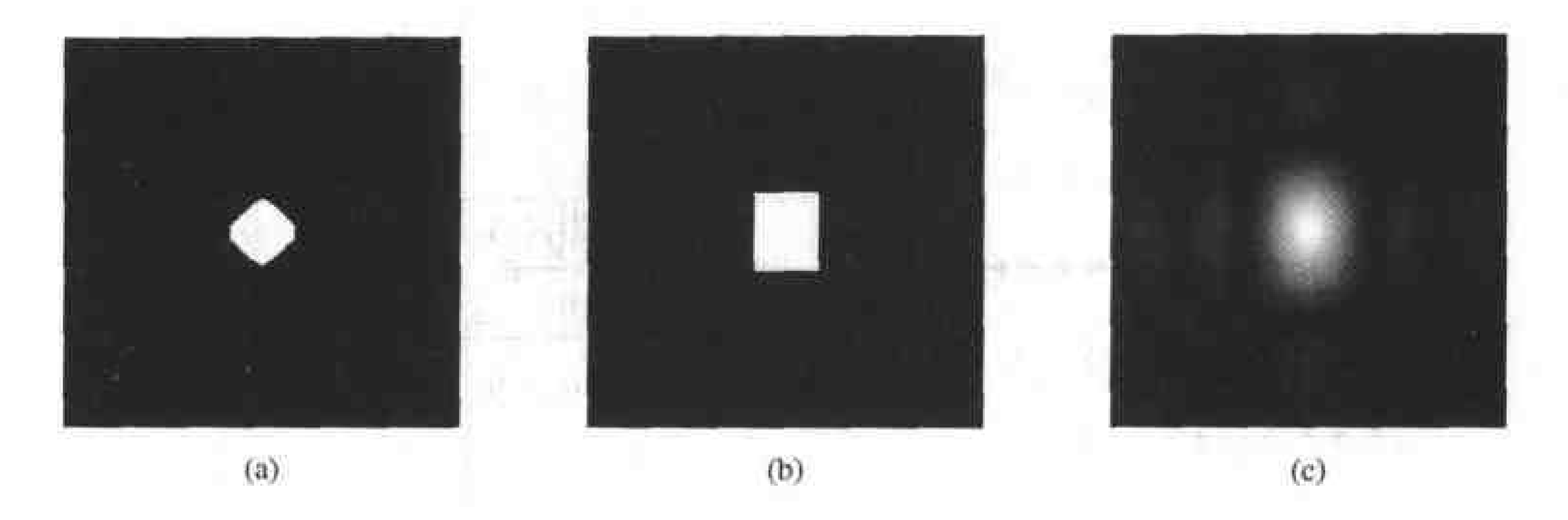


图 4.11 从 PAL 到 NTSC 转换滤波器的频率响应：(a)理想滤波器；(b)相应于图 4.9 中的 H_2 和 H_3 级联的复合滤波器，这里 H_2 和 H_3 都是理想滤波器；(c)当用公式(4.2.15和4.2.16)实现行率下转换(H_2)和帧率上转换(H_3)时的复合滤波器。图(a)考虑了去隔行和隔行滤波器，而(b)和(c)未考虑。在每个图中，垂直轴代表 f_y，水平轴代表 f_t。尽管滤波器的响应定义于 $|f_y|\leqslant 12.5/\Delta_y$，$|f_t|\leqslant 2.5/\Delta_t$，但只画出了由 $|f_y|\leqslant 2.5/\Delta_y$，$|f_t|\leqslant 2.5/\Delta_t$ 确定的频率区

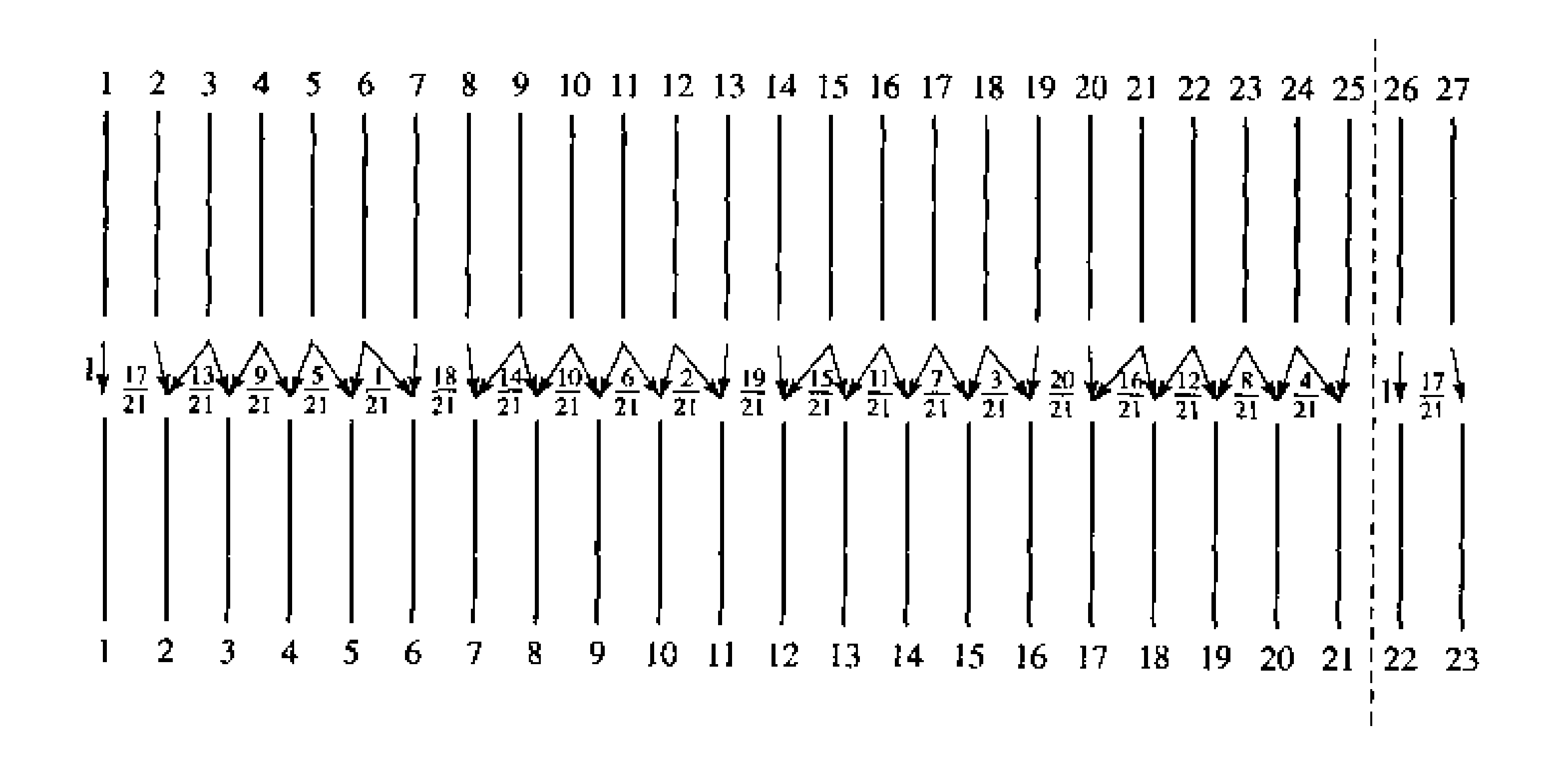


图 4.12 通过将每 25 行转换成 21 行，并使用两个最接近的已知行(图中上面的一行)内插出每一个新的行(图中下面的一行)[5]，实现从625行到525行的下转换。对图中下面的每一行，图中给出了与上面一行中左边相邻行有关的内插系数。右边相邻行的内插系数是1减去左边的系数

这是一个用于由 625 行内插成 13 125 行或者在每两行间插入 21 行的线性内插滤波器。但是如果只使用公式(4.2.15)所给出的 21 个可能的内插系数中的两个，则最多有一行保留在 525 行格栅中。

为了实现从 50 Hz 到 60 Hz 的时间转换，这里不使用首先将帧率从 50 fps 向上转换到 300 fps，然后再向下采样到 60 fps 的方法，一个简单得多的方法是把每 5 帧转换为 6 帧，如图 4.13[13]所示。正如图中所示，6 个输出帧的每一帧都是用 5 个输入帧中的两帧内插得到，在这里与每个输入帧有关的内插系数是与输出帧和输入帧之间的距离成反比的。按照前面描述的行率转换步骤，我们可以确定出内插系数，示于图 4.13 中。在 300 帧格栅上的等效时间滤波器是：

$$h_t(t_k)=\begin{cases}(1-|k|/6), & t_k=k\Delta_{t,3}, |k|=0,1,\cdots,5\\ 0 & \text{其他}\end{cases} \tag{4.2.16}$$

复合时空滤波器的频率响应示于图 4.11(c)。与示于图 4.11(b)的所需响应相比,我们看到,线性内插导致模糊的过渡以及阻带中的许多振铃。

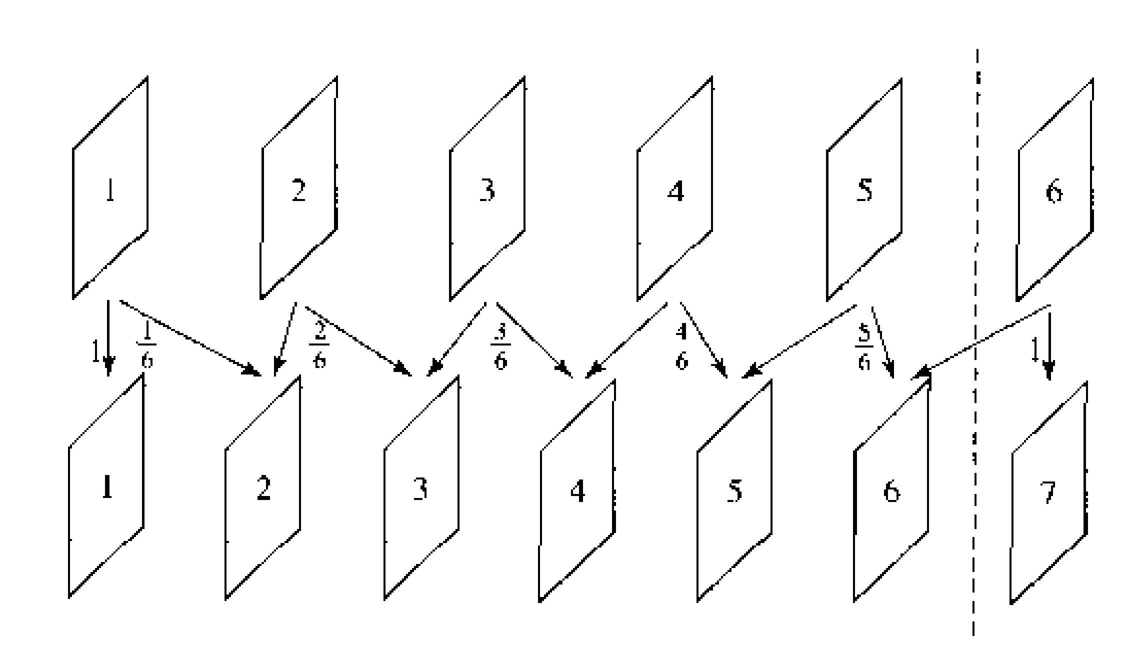


图 4.13 通过将每 5 帧转换为 6 帧,以及由两个最近的已知帧内插每一个新帧,实现从 50 场到60场的上转换。对于图中下面一行的每一帧,与图中上面一行中左边相邻帧有关的内插系数已在图中给出了,右边相邻帧的内插系数是1减去左边的内插系数

4.2.3 运动自适应内插

从前面两小节给出的例子可以看到,在空间和时间方向上使用内插各有其优点和缺点。对于静止的区域,时间内插将产生精确的结果,它确实提高了给定帧的分辨率。另一方面,在相继的场或帧间含有快速时间变化的区域中,两个分离的场或帧中具有相同空间标号的像素可能对应于景物中不同的目标区域。在这种情况下,时间内插将产生错误的结果,而单独的空间内插可能会好一些。为了克服这些问题,可以采用运动自适应内插滤波器。这种滤波器是根据一个运动检测器的输出,对于每一个像素在垂直和时间内插之间进行切换。若检测到运动,则只使用垂直内插;否则,只使用时间内插。可以基于每个像素周围一个小邻域内的时间变化进行该像素的运动检测。这个方法成功的关键在于运动检测过程的精确性。

除使用空间和时间内插间的硬切换之外,我们也可以使用二者加权的平均,可以通过测量被内插像素周围一个小邻域内时间和空间的变化来确定权重。令 σ_s^2 和 σ_t^2 分别代表空间和时间变化的大小,$\hat{\psi}_s$ 和 $\hat{\psi}_t$ 分别代表用空间内插和时间内插的内插值。最终的结果将是:

$$\hat{\psi}=w_t\hat{\psi}_t+w_s\hat{\psi}_s, \qquad w_t=\frac{\sigma_s^2}{\sigma_s^2+\sigma_t^2}, w_s=\frac{\sigma_t^2}{\sigma_s^2+\sigma_t^2} \tag{4.2.17}$$

为了进一步改进含有运动的区域的性能,对运动补偿内插也已经进行了研究。使用这种方法时,如果一个像素被认为落到了一个运动区域,那么它在相邻场中的对应位置就被确定了,而且用于提供时间内插的结果,即公式(4.2.17)中的 $\hat{\psi}_t$。最具有挑战性的问题是运动估计,即在相邻的场或帧中寻找对应的点。我们将在第 6 章详细讨论运动估计问题。为了更深入研究运动补偿采样率转换,读者可以参阅参考文献[2,3]。另一种方法是根据空间频率的内

容改变采样格栅[1]。

4.3 小结

采样率转换的理想解决方案(4.1节)

- 图4.1示出了不同转换问题理想的解决方案。公式(4.1.4,4.1.9,4.1.15)给出了理想的滤波器。上采样问题是一般转换问题在 $\Lambda_3=\Lambda_2$ 时的特例,下采样问题是 $\Lambda_3=\Lambda_1$ 时的特例。
- 在一般的速率转换问题中,不必在 Λ_3 中进行滤波运算。正如公式(4.1.20)指出的,为了内插 Λ_2 中的每个采样值,只需对 Λ_1 中的采样值进行滤波。

视频采样率转换的实际解决方案(4.2节)

- 通常视频速率转换问题包括:去隔行;NTSC与PAL之间的转换,或者等效于BT.601 525/60与625/50之间的转换;SDTV与HDTV之间的转换;BT.601 4:2:2与4:2:0之间的转换;电影胶片与NTSC或PAL之间的转换。
- 在实践中,一般用很短的空间和时间滤波器,以便降低存储和计算的需求。
- 重要的是确定理想的解决方案,并用它作为准则评价实际的方案。
- 运动自适应运算对于时间方向上的滤波是重要的。其性能取决于运动检测和估计的精度。

4.4 习题

4.1 在4.2.1节中,我们讨论了几种用于去隔行的实际滤波器。我们看到,行-场平均法最接近理想的滤波器;然而,它需要存储两帧。为了减少帧存储量,已经提出了一种单帧内插滤波器[5]。这种方法在同一场中使用垂直内插,由同一帧的另外一场进行时间内插。在图4.5中,场 t 中的行 D 用 $D=(C+E)/2+K/4-(I+M)/8$ 估计。请确定在去隔行点阵中的等效滤波器及其频率响应。画出幅度响应(用MATLAB),然后将它与图4.7相比较。这个滤波器具有实频率响应吗?如果不具有,为什么?

4.2 考虑下面的去隔行方法:

$$\hat{f}(t,m)=\frac{1}{2}f(t-1,m)+\frac{9}{32}[f(t,m-1)+f(t,m+1)]$$
$$-\frac{1}{32}[f(t,m-3)+f(t,m+3)]$$

这里 $f(t,m)$ 代表第 t 场和第 m 行的图像值。对于第 t 场,我们假设行 $m+2k,k=0,1,\ldots$ 丢失。根据所给出的运算求等效的内插滤波器并计算它的频率响应。用MATLAB画出该频率响应,并将它与用于去隔行的理想内插滤波器进行比较。

4.3 对于如4.2.2节讨论的直接和顺序的PAL到NTSC转换方法,比较它们的计算需求。假设在一个总共有 N 个样点的区域中滤波需要进行 $\alpha N\log_2 N$ 次运算。比较两种方

法的总运算量。只需统计涉及到滤波的运算;涉及向上和向下转换的运算量微小,可以忽略。

4.4　考虑示于图4.14的用于NTSC(525 lpf,30 fps,隔行)到PAL(625 lpf,25 fps,隔行)转换的两种方法。

(a) 对于每一种方法,画出在每个中间步骤中的采样点阵及其反商点阵,并确定所涉及的滤波器的所需频率响应。为简化起见,只在时间-垂直平面上进行。当在频率域画点阵和沃罗纳区域时,假定时间和空间频率轴已被定标,使得 $\frac{1}{\Delta t}\text{cps} = \frac{1}{\Delta y}$ 周/图像高度,对于 $\Delta t = \frac{1}{60}\text{s}, \Delta y = \frac{1}{525}$ 图像高度。

(b) 顺序实现可以组合,从而可以像直接方法那样用一个运算实现。求总的等效滤波器响应并把它与直接方法的滤波器进行比较。

(c) 采用与习题4.3同样的假设,比较这两种方法总的运算量。

(d) 根据你对(b)和(c)的答案,评论一下这两种方法的优缺点。

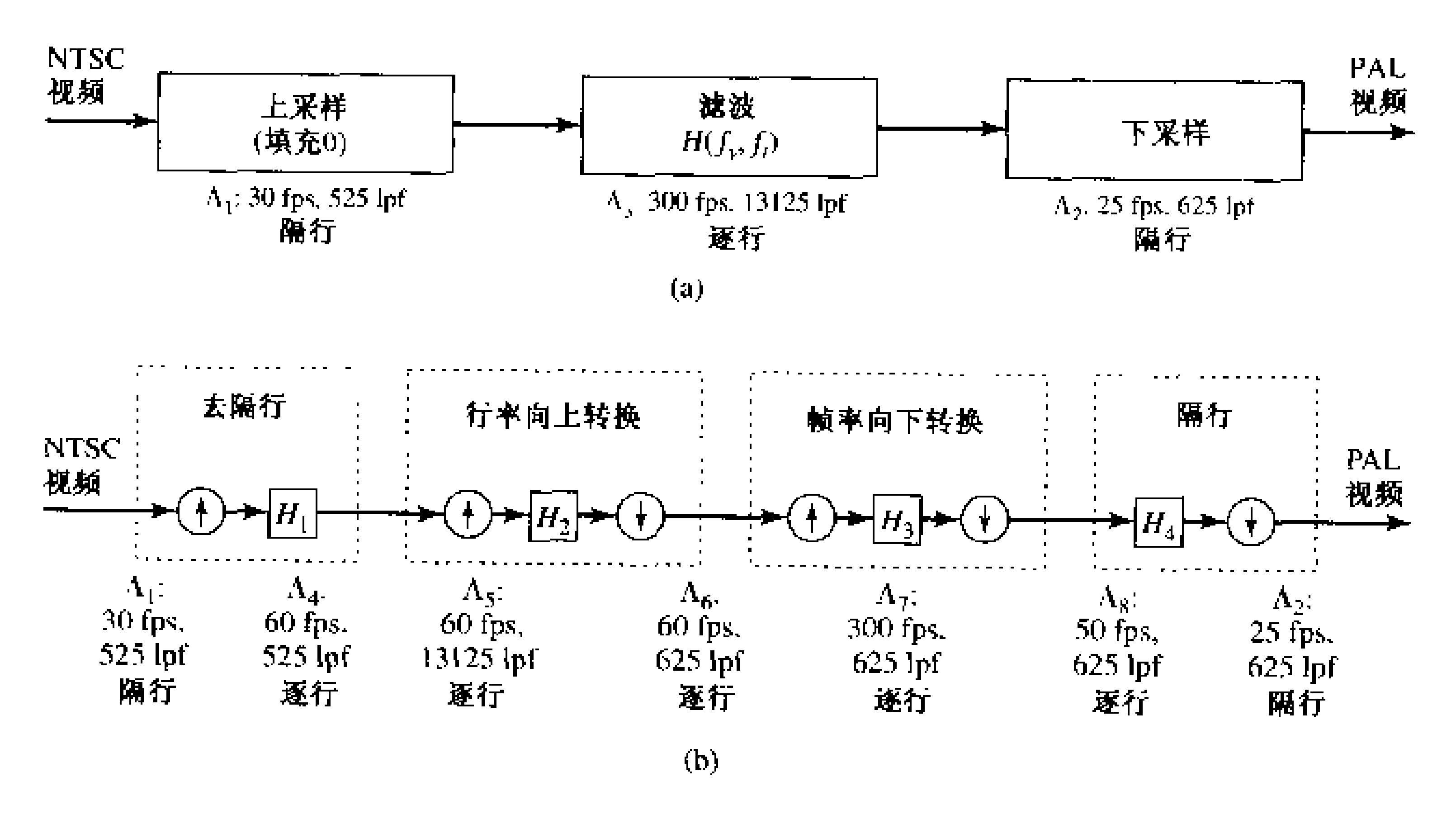


图4.14　用于NTSC到PAL制式转换的两种不同方法
(a)直接转换 (b)顺序实现

4.5　考虑将一个以24 fps逐行摄取的电影胶片转换到具有30 fps隔行的NTSC电视制式的问题。

(a) 画出对应这两种采样格式的点阵。确定转换的中间点阵,以及在这个点阵上的理想滤波器。

(b) 实际上,通常使用如图4.15所示的简单的3:2下拉运算。我们可以看到,有一半的帧被转化成三场,而另一半的帧被转化成两场。为了实现简单,未使用任何滤波器。你能求出在(a)中确定的中间点阵的等效滤波器吗?

(c) 你能提出其他比(b)更好的简单方法吗?

4.6　考虑将BT.601 4:2:2信号转换成4:2:0的问题(请记住,在这两种格式中,Y分量是相同的,而4:2:0格式中Cb和Cr分量所具有的行数是4:2:2格式中的一半)。由于

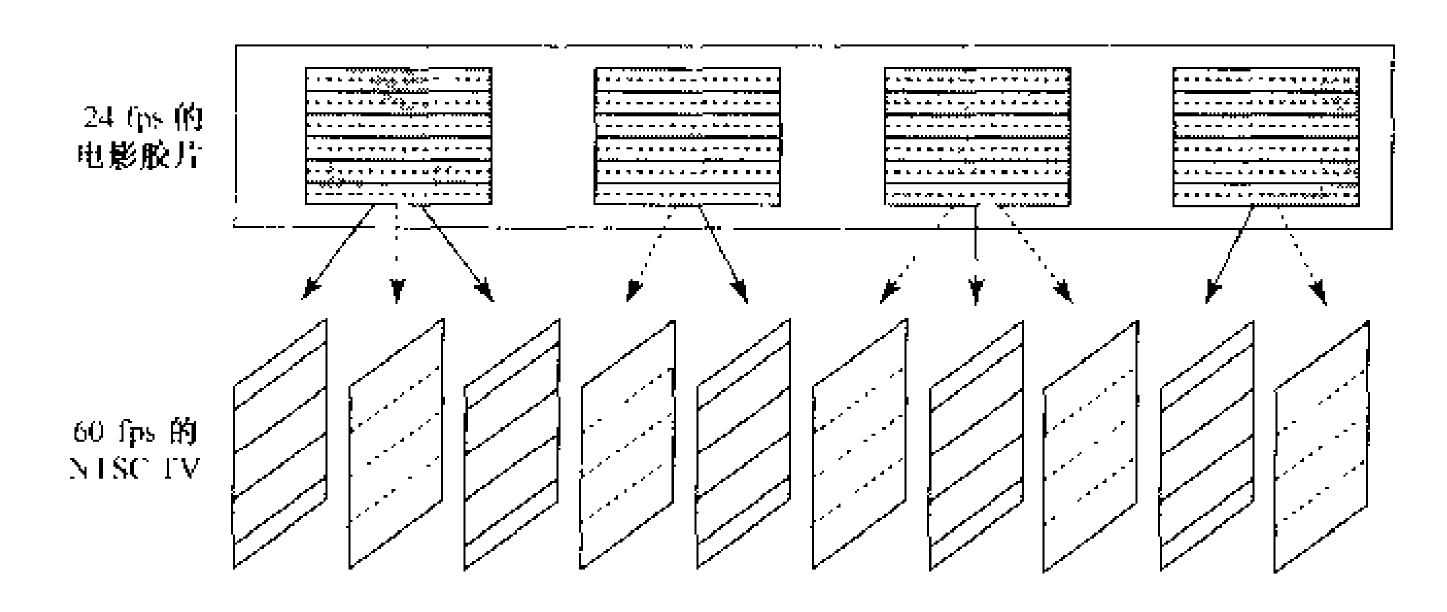


图 4.15 用于将 24 fps 电影胶片转换成 60 fps 的 NTSC TV 的 3:2 下拉运算[5]

两种格式都是隔行的，所以这不是一个简单的 2:1 下采样的问题。图 4.16 画出了色度分量的转换问题。

(a) 画图说明 BT.601 4:2:2 和 4:2:0 格式的采样点阵，并确定中间点阵及其理想的内插滤波器。

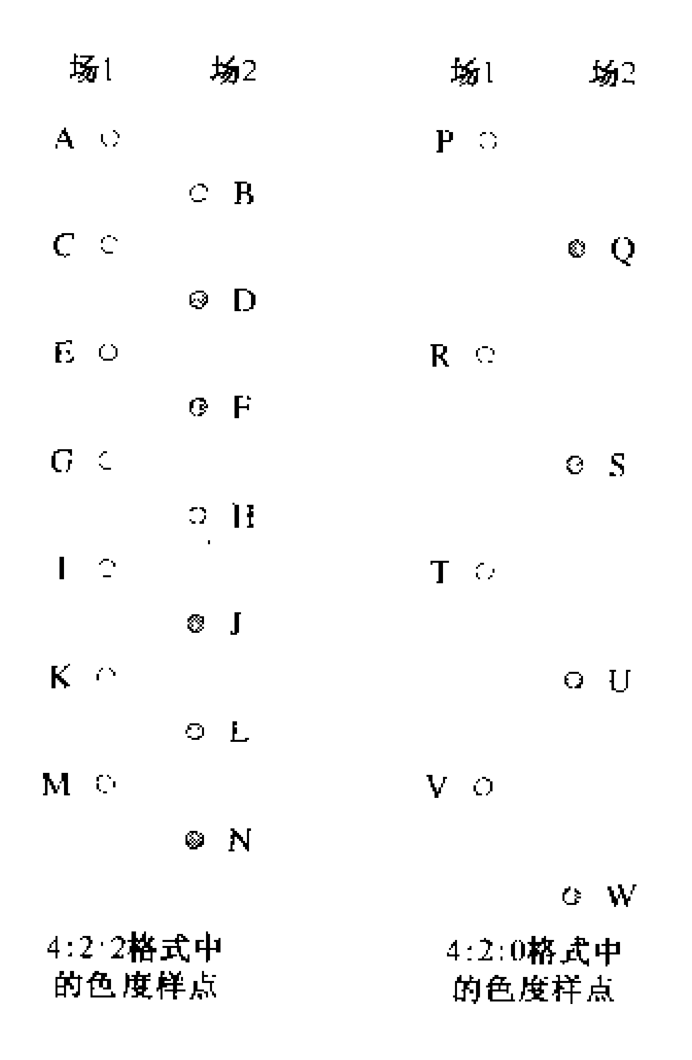


图 4.16 BT.601 4:2:2 和 4:2:0 格式中的色度采样点之间的关系

(b) 为了简单起见，我们可以将滤波限制在同一场中，即场 1(场 2)的采样点只用在场 1(场 2)中的滤波进行下采样。实际上这是在 MPEG-2 标准中建议的把4:2:2格式转换成4:2:0 的一种技术。在这种情况下，场 1 必须进行 2 倍的下采样。所建议的预滤波运算是：

$$G' = (-29A + 88E + 138G + 88I - 29M)/138$$

对于场 2，必须进行 2 倍的内插来获得所需的采样点。所建议的内插运算是：

$$S = (D + 7F + 7H + J)/16$$

对于上述重采样方案,确定场 1 和场 2 的等效垂直滤波器,以及总的垂直-时间滤波器。将结果与(a)的理想滤波器进行比较。

4.7 习题 4.6 考虑了从 4:2:2 到 4:2:0 格式的转换。这里我们考虑从 4:2:0 到 4:2:2 格式的转换问题。

(a) 画出中间点阵并确定在这个点阵上的理想内插滤波器。

(b) 对于如下的实际方案确定等效滤波器及其频率响应[5]:对于场 1,需要进行 2 倍的内插,用两个相邻样点的平均来获得丢失的样点:

$$G = (R + T)/2$$

对于场 2,需要进行 4 倍的内插,后跟一个 2 倍的下采样。最终的运算是:

$$D = (3Q + S)/4, \qquad F = (Q + 3S)/4$$

确定场 1 和场 2 的等效垂直滤波器,以及在内插格栅上总的垂直-时间滤波器。求出它们的频率响应。与(a)中的结果进行比较。

4.5 文献目录

[1] Belfor, R. A. F., M. P. A. Hesp, and R. L. Lagendijk. Spatially adaptive subsampling of image sequences. *IEEE Trans. Image Process.* (Sept. 1994), 3(5):492–500.

[2] Belfor, R. A. F., R. L. Lagendijk, and J. Biemond. Subsampling of digital image sequences using motion information. In M. I. Sezan and R. L. Lagendijk, eds., *Motion Analysis and Image Sequence Processing*. Boston: Kluwer Academic Publishers, 1993, 189–223.

[3] Beuker, R. A., and I. A. Shah. Analysis of interlaced video signals and its applications. *IEEE Trans. Image Process.* (Sept. 1994), 3(5):501–12.

[4] Dudgeon, D. E., and R. M. Mersereau. *Multidimensional Digital Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1984.

[5] Haskell, B. G., A. Puri, and A. N. Netravali. *Digital Video: An Introduction to MPEG-2*. New York: Chapman & Hall, 1997.

[6] Hou, H. S., and H. C. Andrews. Cubic splines for image interpolation and digital filtering. *IEEE Trans. Acoust., Speech, Signal Process.* (Dec. 1978), ASSP-26: 508–17.

[7] Mitra, S. K. *Digital Signal Processing—A Computer-Based Approach*. New York: McGraw Hill, 1998.

第5章 视频模型

数字视频图像处理的任务之一是描述一个视频序列中相继图像之间在物体的运动和其他效果方面的变化，例如照度的改变和摄像机运动。为了把真实世界的变化与视频序列的变化联系起来，我们需要描述真实世界和图像生成过程的参数化模型。最重要的模型是场景、物体、摄像机以及照度模型。这些模型描述了我们对真实世界做出的一些假设。依赖于所选定的模型，我们可以用或多或少的精度和细节来描述真实世界。使用图像分析工具，可以从真实世界图像的视频序列中估计出参数模型的参数。用这些参数模型和估计出的参数，我们可以重建一个近似于真实世界的模型世界。表5.1示出了我们用来命名参数模型的一些术语、它们相应的真实世界实体，以及根据参数模型重建的实体和在模型世界中给出的实体。我们用限定词“真实”和“模型”来区别真实世界和模型世界的术语。当上下文很明确的时候，我们可能忽略这个限定词。

在下面各节中，我们描述摄像机模型(5.1节)，照度模型(5.2节)，物体模型(5.3节)以及场景模型(5.4节)。目前大多数视频处理应用中的物体模型都假定物体做二维运动。我们在5.5节中描述有关的二维运动模型。

取决于所使用的模型，模型参数估计可能是困难的。在第6章和第7章中，我们将分别讨论二维和三维运动的模型参数估计问题。

表5.1 真实世界和模型实体的命名约定

真实世界	参数模型	模型世界
真实景物	场景模型	模型场景
真实物体	物体模型	模型物体
真实纹理	纹理模型	模型纹理
真实形状	形状模型	模型形状
真实运动	运动模型	模型运动
真实摄像机	摄像机模型	模型摄像机
真实图像	图像模型	模型图像
真实照明	照度模型	模型照度

5.1 摄像机模型

摄像机模型描述真实场景中的真实物体在真实摄像机成像平面上的投影。成像平面也称为摄像机靶。然后把成像平面上的图像转换为数字图像，如第1章所述。

5.1.1 针孔模型

真实物体在真实摄像机靶上投影的一种广泛使用的近似是针孔摄像机模型，如图5.1所

示。在这幅图中，F 代表摄像机的焦距，$\mathbf{C}$ 代表焦点（也称为摄像机中心）。一个三维点 $\mathbf{X}$ 的投影位置 $\mathbf{x}$ 是连接 $\mathbf{X}$ 和 $\mathbf{C}$ 的直线与成像平面的交点。因为成像平面位于焦点之后，所以成像位置与它真实的三维位置是相反的。大多数在视频处理中采用的摄像机模型把成像平面和物体放在焦点的同侧，如图 5.2(a)所示，以避免处理成像平面中被反转的位置。

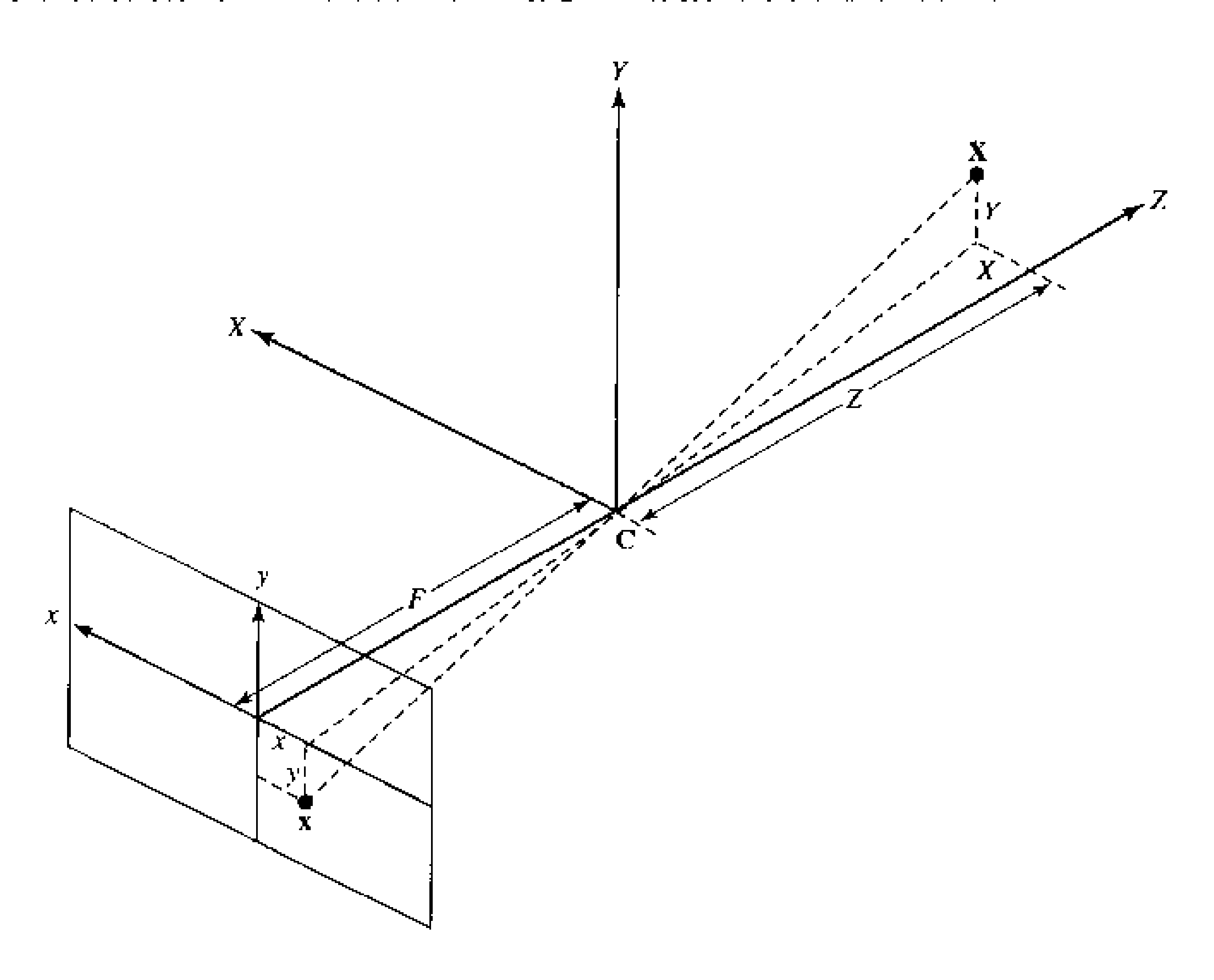


图 5.1 针孔摄像机的透视投影

不失一般性，我们假设（见图 5.2(a)）三维坐标系统（也称为世界坐标系统）的原点位于焦点，它的 XY 平面与成像平面平行，而且世界坐标$(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{Z})$遵循右手系统，以 Z 轴的正方向为成像方向。进一步，我们假设成像平面使用与三维坐标同样的距离单位。从图 5.2(a)所示的相似三角形，我们有：

$$\frac{x}{F}=\frac{X}{Z}, \qquad \frac{y}{F}=\frac{Y}{Z}, \tag{5.1.1}$$

或

$$x=F\,\frac{X}{Z}, \qquad y=F\,\frac{Y}{Z} \tag{5.1.2}$$

这个关系称为透视投影。透视投影的一个显著特性是物体离摄像机越远它所成的图像就越小。在数学上，这可以描述为投影的 x 和 y 值与深度 Z 的值成反比关系。

当成像物体距离摄像机平面很远时，透视投影可以用正交投影近似，也称为平行投影（见图 5.2(b)）：

$$x=X, \qquad y=Y \tag{5.1.3}$$

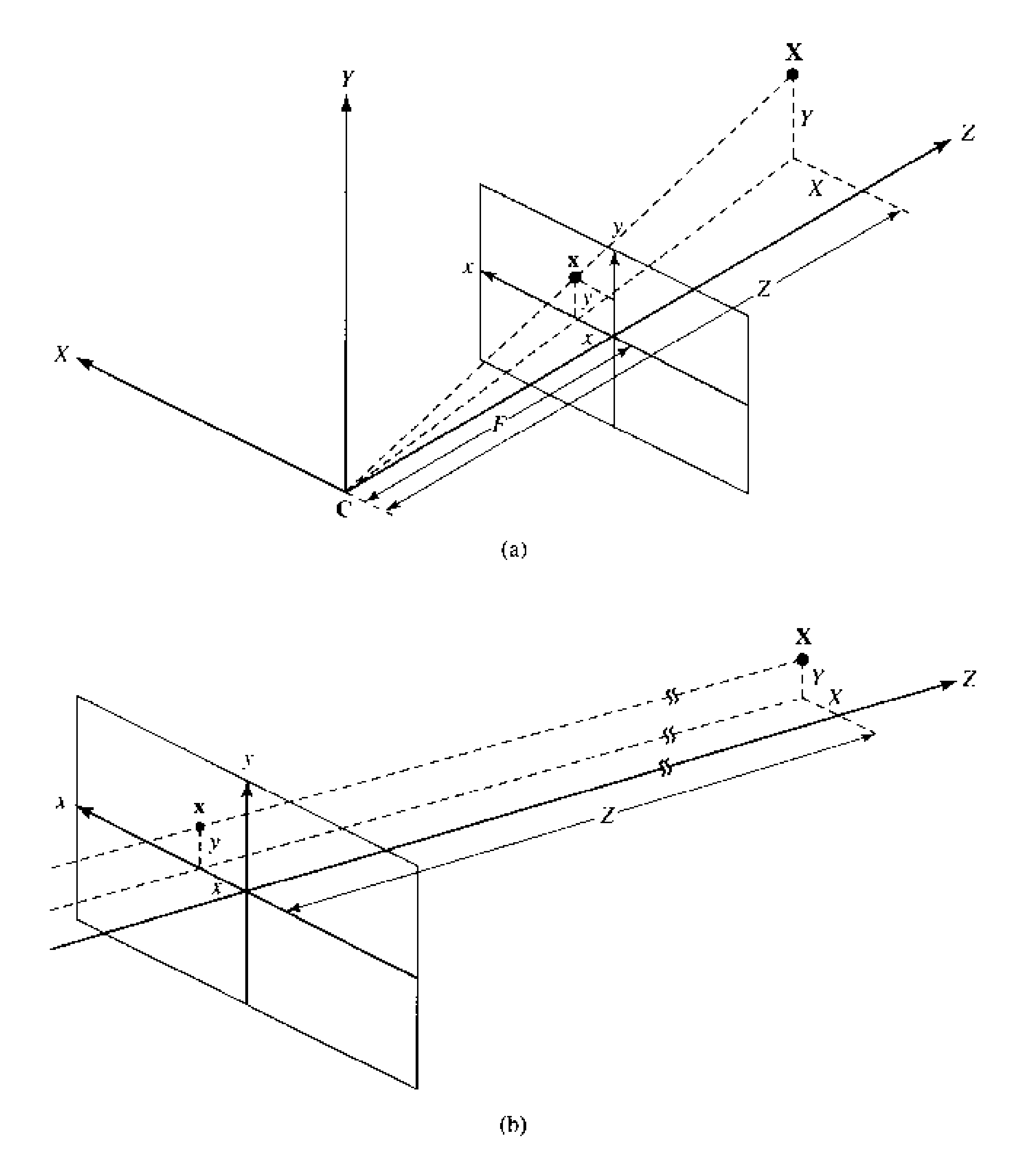


图 5.2 摄像机模型:(a)在针孔摄像机中使用透视投影;(b)用平行投影作为针孔摄像机的近似。在这两种情况之下,假设世界坐标系统的 XY 平面与图像平面是完全对准的

显然,这个关系是非常简单的,而且使三维和二维之间的变换大大简化。实际上,物体必须距离摄像机很远的要求可以放宽。只要物体表面相对深度的变化与物体到摄像机的距离相比可以忽略时,就可以可靠地使用这个近似。

注意,透视投影和正交投影都是多对一的映射。实际上,在连接焦点 **C** 和图像点 **x** 的光线上的所有三维点都会以 **x** 作为它们的投影点。我们把这条光线命名为观测线。在成像平面上看到的只是这条光线所遇到的第一个物体点。摄像机模型中所用的这种投影的多对一映射,使基于二维图像估计物体的三维结构和运动成为非常困难和具有挑战性的问题。

具有透视投影的针孔摄像机模型只是大多数真实摄像机的一种近似。它没有考虑摄像机轴与图像中心对不准、真实镜头有限孔径的低通滤波作用、有限曝光时间以及镜头的其他失真问题。

5.1.2 CAHV模型

在一些应用中，我们希望对摄像机的描述能够使我们容纳摄像机的运动。进一步，我们希望校准摄像机以补偿针孔摄像机模型与真实摄像机之间的几何误差。CAHV摄像机模型[16]用四个矢量描述针孔摄像机模型的透视投影：

C：指向摄像机中心的矢量；

A：在光轴方向上的单位矢量；

$\mathbf{H}_0$：成像平面水平轴方向上的单位矢量；

$\mathbf{V}_0$：成像平面垂直轴方向上的单位矢量。

图5.3画出了这个几何图形。改写公式(5.1.2)的透视投影，把矢量(**P** - **C**)投影到摄像机轴**A**以及成像平面轴$\mathbf{H}_0$和$\mathbf{V}_0$，按下式把点**P**投影到图像点**p**：

$$\mathbf{p}=\begin{pmatrix}x\\y\end{pmatrix}=\frac{F}{\mathbf{A}^T\cdot(\mathbf{P}-\mathbf{C})}\cdot\begin{pmatrix}\mathbf{H}_0^T\cdot(\mathbf{P}-\mathbf{C})\\\mathbf{V}_0^T\cdot(\mathbf{P}-\mathbf{C})\end{pmatrix}\tag{5.1.4}$$

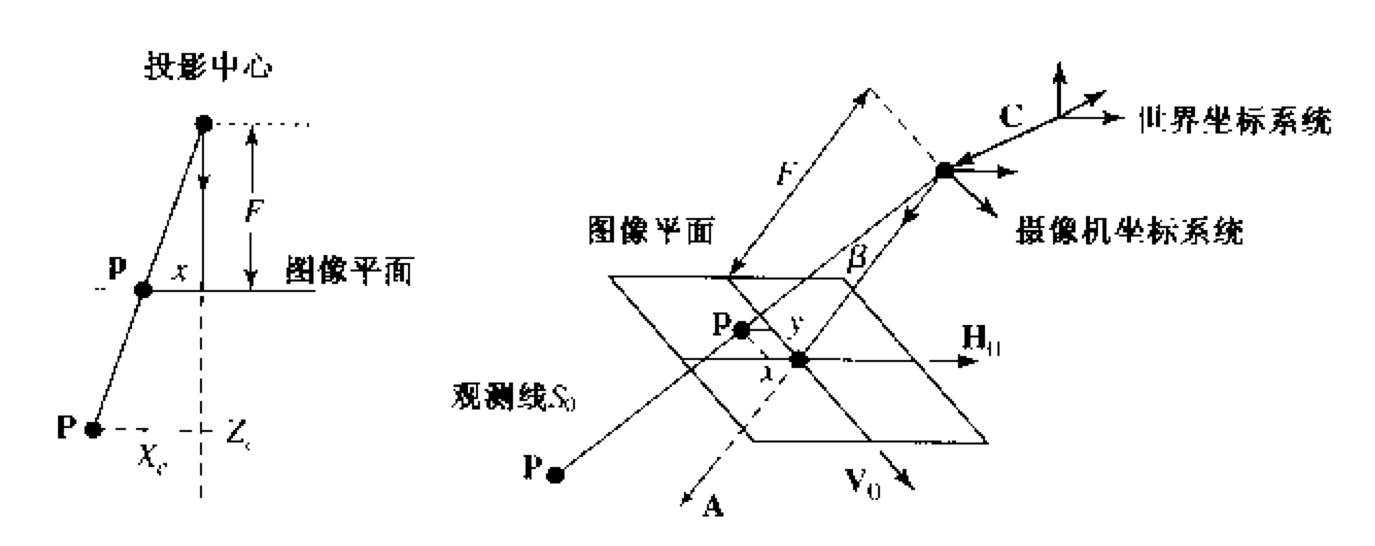


图5.3 用CAHV摄像机模型得到的空间一点**P**到成像平面上一点**p**的透视投影

实际上，图像平面坐标和世界坐标并不总是如针孔摄像机模型按照公式(5.1.2)所假设的那样完全对准的。CAHV摄像机模型通过它的非固有参数**C**和**A**以及它的固有参数$\mathbf{H}_0$，$\mathbf{V}_0$和F，可以表征实际的摄像机系统。这些参数使我们能够描述一个偏离摄像机轴的成像平面，也可以描述由光学系统引入的某些失真。附加的固有摄像机参数描述镜头的径向失真以及镜头的调制转移函数(MFT)。MFT可以用一个低通滤波器模型化。可以用摄像机校准算法估算这些参数；有兴趣的读者可以参阅参考文献[2]。当由视频序列估计三维形状和运动时，准确地了解摄像机参数是尤其有用的。最近，研究人员提出了不需校准摄像机而从一个图像序列提取三维形状和运动的算法[10]。

5.1.3 摄像机运动

图5.4示出了典型的摄像机运动。摄像机沿着成像平面的水平轴($\mathbf{H}_0$)和垂直轴($\mathbf{V}_0$)平移分别称为跟(track)和吊(boom)。沿摄像机光轴方向上的平移被称为推(dolly)。摇(pan)指的是绕垂直轴($\mathbf{V}_0$)旋转，而倾(tilt)指的是绕水平轴($\mathbf{H}_0$)旋转。摄像机绕光轴旋转称为滚(roll)。除了具有这些运动类型之外，摄像机还能够改变它的焦距，称为变焦。正如前面提到的，CAHV摄像机模型很适合于表示这些各种各样的摄像机运动。

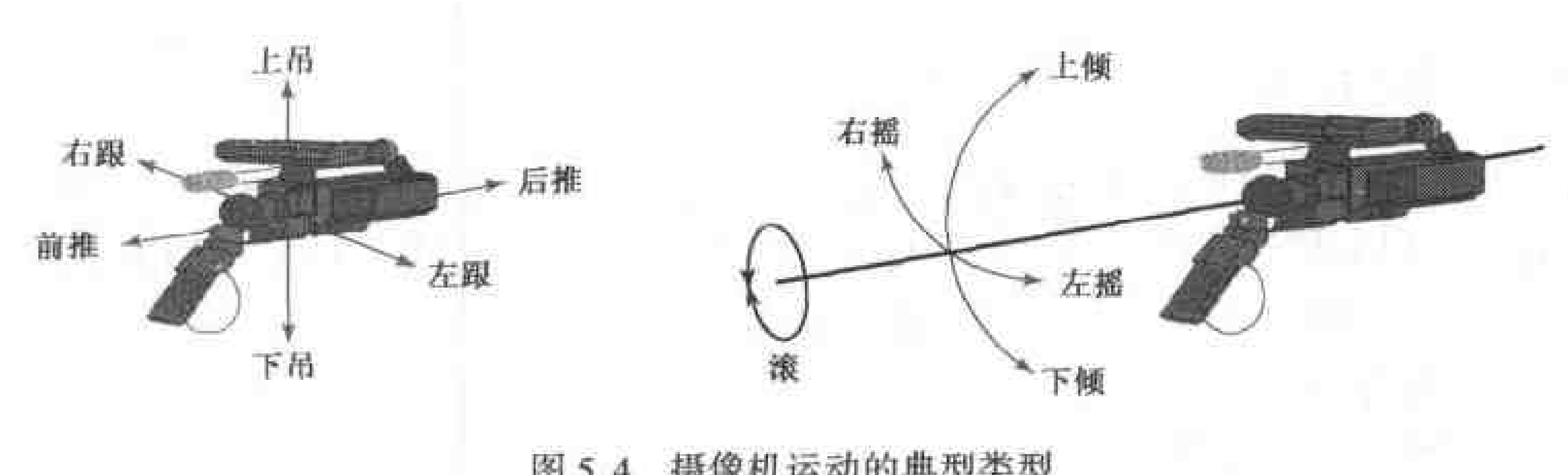


图 5.4 摄像机运动的典型类型

5.2 照明模型

正如 1.1.1 节描述的那样,有两种类型的光源:照明和反射。照明模型描述入射到物体上的光如何影响反射光的分布,我们所看到的就是这种分布。在这一节中,我们描述几种这样的模型。照明模型可以分为光谱模型和几何模型。如果我们想要对来自几种彩色光源,或由不同色彩的物体或彩色反射面间接照明的彩色的变化建模,可以使用光谱照明模型。几何模型描述入射光的幅度和方向分布。在 1.1.1 节中已经讨论了光谱的特性,本节讨论几何模型。

几何照明模型适用于环境和点光源。对于每一种类型的光源,我们也可以把它区分为局部和总体照明模型。局部照明模型假设照明与物体的位置以及其他物体无关。总体照明模型还要考虑物体间的影响,例如阴影以及从物体表面反射的光。

5.2.1 漫反射和镜面反射

我们只能看见反光的物体,而所感觉的颜色取决于它们反射的波长范围。一般地,反射可以被分解为两个分量:漫反射在所有方向上具有相等的能量分布;镜面反射在入射光的镜向上强度最大(关于漫反射和镜面反射的很好的说明参见参考文献[3])。只呈现漫反射的表面称为朗伯表面,更通常地描述为钝的或无光泽的——木头表面和水泥墙属于这一类。由于漫反射,我们才能感觉到物体的颜色。镜面反射可以用发亮的表面和镜子观察到。镜面反射不能显示物体的颜色,只显示入射光的颜色;因此,我们实际上不能感觉到一个表面上只呈现镜面反射的物体的颜色。除了镜子以外,通常的表面既具有漫反射也具有镜面反射。只有漫反射决定物体表面的颜色。在日光下看一个人的眼睛,除了一个白点之外,我们通常看到的是眼睛本身,这个白点是由太阳光的镜面反射产生的。根据经验可知,如果我们相对于眼睛改变我们的位置,那么这个点会在眼睛表面移动。

5.2.2 在不同照明和反射条件下的光强分布

在视频处理中,照明模型主要用于描述由真实世界的照明变化引起的视频序列时间上的变化。由于物体与它所投射的影子一起运动,背景的照明可能发生变化。由于物体表面反射光线,这个反射源会改变场景的整体照明。

当讨论对于一个物体表面光源的相互作用时,涉及到三种类型的能量。第一个是入射光通量,它指的是从光源发射出的能量的速率,用瓦特(W)度量。第二个是入射辐照度,它指的是物体上每单位表面积的入射光通量,单位是 W/m^2(注意,物体上某一点的辐照度取决于入射

光与表面在该点的法线之间的夹角)。最后一个是反射辐射强度,它度量从物体表面反射的光的能量。

反射辐射强度 C 的分布取决于入射光辐照度 E 的分布和物体表面在该点的反射系数函数 r。最一般的关系可以描述为:

$$C(\mathbf{L},\mathbf{V},\mathbf{N},\mathbf{X},t,\lambda)=r(\mathbf{L},\mathbf{V},\mathbf{N},\mathbf{X},t,\lambda)\cdot E(\mathbf{L},\mathbf{N},\mathbf{X},t,\lambda) \tag{5.2.1}$$

其中 $\mathbf{X}$ 是物体表面的位置,$\mathbf{N}$ 是在位置 $\mathbf{X}$ 处的表面法线矢量,$\mathbf{L}$ 是照明方向,$\mathbf{V}$ 是连接 $\mathbf{X}$ 与摄像机焦点的观测方向,λ 是光的波长(见图5.5)[12]。显然,$\mathbf{L}$,$\mathbf{V}$ 和 $\mathbf{N}$ 是 $\mathbf{X}$ 和 t 的函数。反射系数函数 r 定义为反射光强度(即光通亮)与入射光强度之间的比值。这个标量函数 r 也称为漫反射系数,或者简称为反射系数。反射系数函数取决于入射光的波长、表面的几何形状和材料质地。当物体移动时,同一位置上的反射系数随时间变化。注意,$r(\mathbf{L},\mathbf{V},\mathbf{N},\mathbf{X},t,\lambda)$ 只是对属于时刻 t 时表面上那些 $\mathbf{X}$ 定义的。

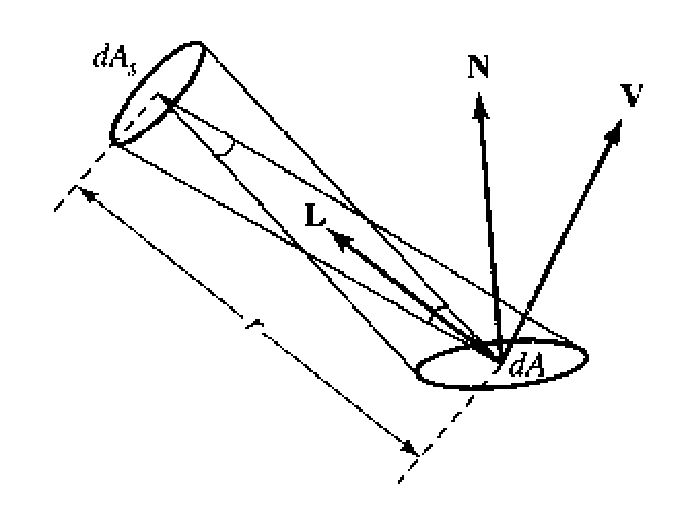


图5.5　从方向 L 被一个具有无限小区域 dA_s 的点光源照射的一个小块表面 dA(具有法线矢量N)。这个小块是从方向V观测的

在下面的段落中,我们引入几个简化的假设,以便学习更多的关于反射辐射强度(公式5.2.1)的知识。我们从假设不透明的物体表面和时不变的照明(以及观测)方向开始。在这种情况下,公式(5.2.1)简化为:

$$C(\mathbf{N},\mathbf{X},t,\lambda)=r(\mathbf{N},\mathbf{X},t,\lambda)\cdot E(\mathbf{N},\mathbf{X},t,\lambda) \tag{5.2.2}$$

注意,尽管照射和观测方向($\mathbf{V}$ 和 $\mathbf{L}$)是固定的,然而因为物体是运动的,所以入射光辐照度仍然是随时间变化的。

一个环境光源在任何一点的每个方向上辐射相同大小的能量,因此它照射物体时不会产生阴影。当入射光是这样一个环境光源而且物体表面是漫反射时,反射辐射强度的分布是:

$$C(\mathbf{X},t,\lambda)=r(\mathbf{X},t,\lambda)\cdot E_a(t,\lambda) \tag{5.2.3}$$

其中 $E_a(t,\lambda)$ 代表时刻 t 环境光的强度。由于光源是环境光,E_a 不依赖于表面位置 $\mathbf{X}$ 或表面法线 $\mathbf{N}$。由于表面是漫反射的,反射系数函数 r 不依赖于表面法线 $\mathbf{N}$。这个环境光源模型是一种局部照明模型,因为我们无法对阴影之类的总体效果建模。在房间里,被照亮的白色墙壁常常可以模型化为环境光源。在室外,被云彩遮挡的太阳提供环境照明。

现在我们讨论由点光源引起的反射辐射强度。假设光源远离场景,以至于物体的位置对入射光没有影响。对于局部照明模型和漫反射表面,任意物体表面点 $\mathbf{X}$ 处的反射辐射强度取

决于入射光方向 $\mathbf{L}$ 与该点处的表面法线 $\mathbf{N}$ 之间的夹角，记为 θ。显然 $\mathbf{L}^T\mathbf{N}=\cos(\theta)$ 成立。这些量示于图 5.6。令 $E_p(t,\lambda)$ 代表光源的最大辐照度，它是当光垂直于表面时的辐照强度。那么对于处于任意角度 θ 的光，辐照度为 $E_p(t,\lambda)\cos\theta$。从公式(5.2.2)开始，在 $\mathbf{X}$ 处的反射辐射强度简化为[3]：

$$C(\mathbf{X},t,\lambda)=r(\mathbf{X},t,\lambda)\cdot E_p(t,\lambda)\cdot\max(0,\mathbf{L}^T\mathbf{N}) \tag{5.2.4}$$

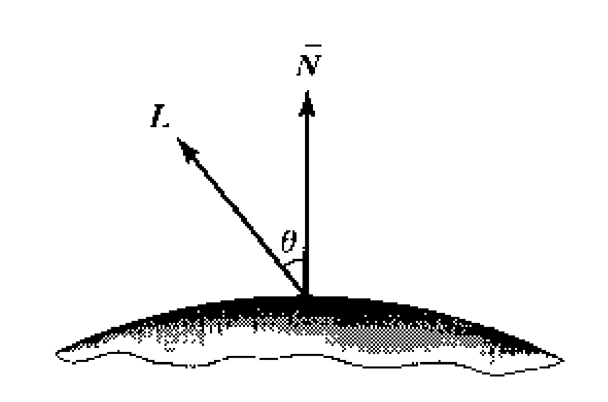


图 5.6 点光源的漫反射在各个方向上是一个常数。强度取决于表面法线$\mathbf{N}$与到光源的方向$\mathbf{L}$之间的夹角θ

在公式(5.2.4)中的 max 运算符是防止物体上那些未接收点光源光线的部分产生负的反射强度。典型的点光源是聚光灯和太阳。当环境光源和点光源都存在时，任意一点总的反射辐射强度是每个光源根据公式(5.2.3 和 5.2.4)计算出的反射辐射强度的叠加。

如果点光源远离物体表面，我们可以把入射光近似为平行光。这个近似对于日光甚至有时是室内光是合法的。在这种情况下，最大入射光辐照度 E_p 不再依赖于表面点 $\mathbf{X}$，而是取决于表面法线 $\mathbf{N}$。假设场景是被一个静止的、远距离的点光源以及一个时间和空间都不变的环境光照明的，则对入射光辐照度的描述可以简化为：

$$E(\mathbf{N},\lambda)=E_a(\lambda)+E_p(\lambda)\cdot\max(0,\mathbf{L}^T\cdot\mathbf{N}) \tag{5.2.5}$$

这是在早期计算机图形学中使用的冯(Phong)阴影模型[11]。假定物体具有各向同性的表面，即 $r(\mathbf{X},\lambda)=r(\lambda)$，相应的反射辐射强度就变成：

$$C(\mathbf{N},\lambda)=r(\lambda)\cdot E(\mathbf{N},\lambda) \tag{5.2.6}$$

5.2.3 物体运动引起的图像函数的变化

现在，我们考察如何用照明模型描述图像平面上的照明效果。假设是白光照明，而且表面法线 $\mathbf{N}$(因此物体的形状)为已知，则公式(5.2.5)中的照明模型可以用来确定由于物体运动引起的一个图像序列中的照明变化。有三个参数需要估计：环境光与点光源辐照度之间的比值 E_a/E_p 以及描述点光源方向 $\mathbf{L}$ 的两个角度。

公式(5.2.5)所给出的模型已经由 Stauder 在基于物体的分析-合成编码器中得到了实现[13]。在二维图像平面中，他假设在时刻 t 像素 $\mathbf{x}$ 的亮度 ψ 正比于在对应于 $\mathbf{x}$ 的三维点 $\mathbf{X}$ 处的反射辐射强度，即：

$$\psi(\mathbf{x},t)=k\cdot C(\mathbf{X})=k\cdot r(\mathbf{X})\cdot E(\mathbf{N}) \tag{5.2.7}$$

其中 k 是常数，$\mathbf{N}$ 表示相应于 $\mathbf{X}$ 的法线方向。这样，从时间 t 到 $t+d_t$，从 $\mathbf{x}_k$ 运动到 $\mathbf{x}_{k+1}$(相应

的表面法线是 $\mathbf{N}_k$ 和 $\mathbf{N}_{k+1}$)的一个点的亮度强度依下式变化：

$$\psi(\mathbf{x}_{k+1}, t+d_t) = \psi(\mathbf{x}_k, t) \cdot \frac{E(\mathbf{N}_{k+1})}{E(\mathbf{N}_k)} \tag{5.2.8}$$

另一种简单的照明模型假设图像信号$\psi(\mathbf{x})$依赖于照度 $E(\mathbf{x})$和反射系数函数 $r(\mathbf{x})$，二者都定义在图像平面上。照度 $E(\mathbf{x})$一般取决于环境光源和点光源。假设环境照明、漫反射表面以及具有正交投影的摄像机模型，则图像信号由反射模型给出：

$$\psi(\mathbf{x}, t) = k_b \cdot E(\mathbf{x}) \cdot r(\mathbf{x}, t) \tag{5.2.9}$$

其中，k_b 是常数。在图像平面中，从 $\mathbf{x}_k$ 运动到 $\mathbf{x}_{k+1}$的一点的信号$\psi(\mathbf{x})$将按下式变化：

$$\psi(\mathbf{x}_{k+1}, t+d_t) = k_b \cdot E(\mathbf{x}_{k+1}) \cdot r(\mathbf{x}_k, t) \tag{5.2.10}$$

因为 $r(x_{k+1}, t+d_t) = r(x_k, t)$成立，即反射系数函数随着物体运动。公式(5.2.10)假设场景的照度没有因物体的运动而改变，即物体没有在场景中投下阴影。已经证明它在使用块匹配的二维运动估计、三维运动估计以及运动物体与其阴影的分割等方面是很有用的[1,4,9,14]。

这种最简单的同时也是应用最广的照明模型简单地假设 $E(\mathbf{x})$是一个常数。在这种情况下，运动点的亮度不变化，公式(5.2.10)简化为：

$$\psi(\mathbf{x}_{k+1}, t+d_t) = \psi(\mathbf{x}_k, t) \tag{5.2.11}$$

这有时称为恒定亮度假设。对于空间和时间不变的环境照明源和漫反射表面，它是合法的。在这种条件下，没有运动的阴影，也没有由光滑表面引起的反射。除非另外说明，本书中所讨论的算法都使用这个假设。

在前面的公式中，我们假设入射光强度是时不变的。然而，只要物体表面不是完全平坦的和静止的，反射辐射强度及亮度图像通常就是随时间和空间变化的。当假设入射光为彩色时，前面的讨论可以推广到视频图像的色度分量。因此，在公式(5.2.7～5.2.11)中，ψ 既可代表亮度函数也可代表彩色函数。

5.3 物体模型

物体模型描述关于真实物体的假设。物体是在一个场景中我们可以认为是分离的项目的实体。对什么是物体的解释可能有所不同，这取决于应用场合。对于交通监视，我们会认为每一辆轿车是一个物体；然而对于视频编码，我们也可能会根据它的运动来定义一个物体，把一辆轿车分割成旋转的车轮和平动的车身。一个物体是用形状、运动和纹理模型描述的[7]。

纹理模型描述一个物体表面的特性。在这里，我们假设物体 m 的纹理用彩色参数 S_m 来描述。这些彩色参数既包含亮度反射系数也包含色度反射系数。由于通常按照公式(5.2.11)假设恒定亮度，因此这些彩色参数对应于图像信号$\psi(\mathbf{x})$。

尽管在本书中所描述的所有物体模型使用相同的纹理模型，然而它们采用不同的运动和形状模型。运动和形状模型通常是从物体的一个近似的三维描述中导出的，以便简化模型参数的估计。在本章中，我们从最一般的形状和运动模型开始。根据需要将介绍对这些模型的简化(见5.5节和第7章的讨论)。

5.3.1 形状模型

一个三维物体的形状由它所占据的三维空间描述。如果我们把三维空间量化成体素，即边长为 w 的小立方体，那么我们就可以对于每一个体素确定它是否属于这个物体。图 5.7 给出了图示，其中 w 表示立方体每个边的长度。物体的形状可以是凸的或凹的，可以有洞。假设物体是拓扑连接的，即我们可以画一条路径从物体上的任何一点到任何其他一点而不离开物体。物体到图像平面上的投影可能会出现拓扑的非连接区域，这是由被遮挡的物体的各部分或位于图像之外的物体的各部分产生的。由于我们对物体的内部并不是很感兴趣，体素表示从存储的角度来说效率不高。因此，我们通常用物体的表面来描述它的形状。时常采用三角形网格，即通常所谓的线框。三角形网格是用称为控制点的顶点 $\mathbf{P}_C^{(i)}$ 构建的。除了这些控制点之外，我们需要一个表来说明哪些控制点定义三角形(见图 5.8)。假设控制点的列表如图 5.8(b)那样被编号，那么我们就可以用这些数字作为控制点的索引。这就把对表面小块或面的定义简化为给出定义这些小块的索引表。这个索引表称为索引面集表，如图 5.8(c)所示。有时，面是由三个以上的点定义的。

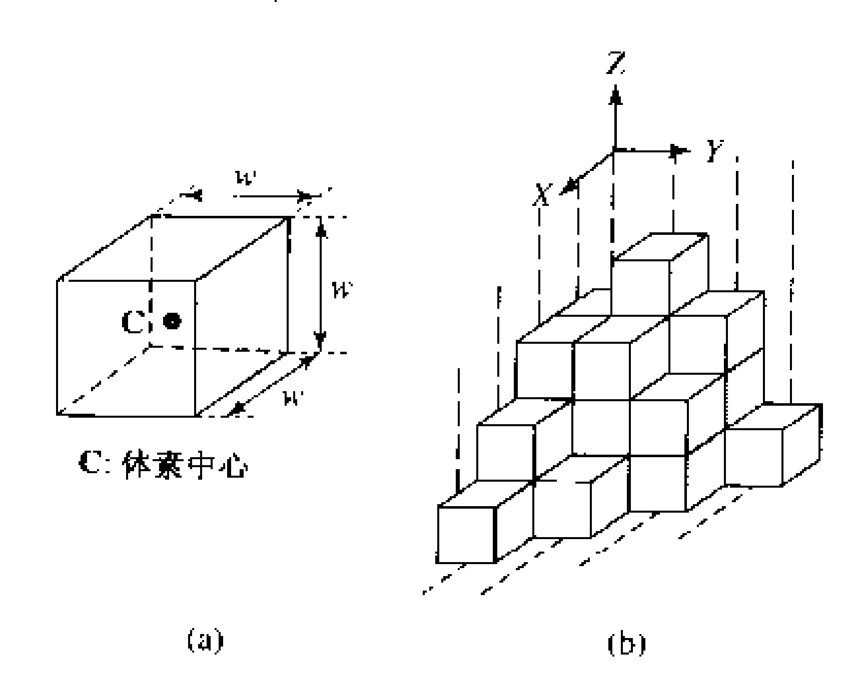


图 5.7 体素：(a)体积元素；(b)在一个三维点阵中用体素表示形状

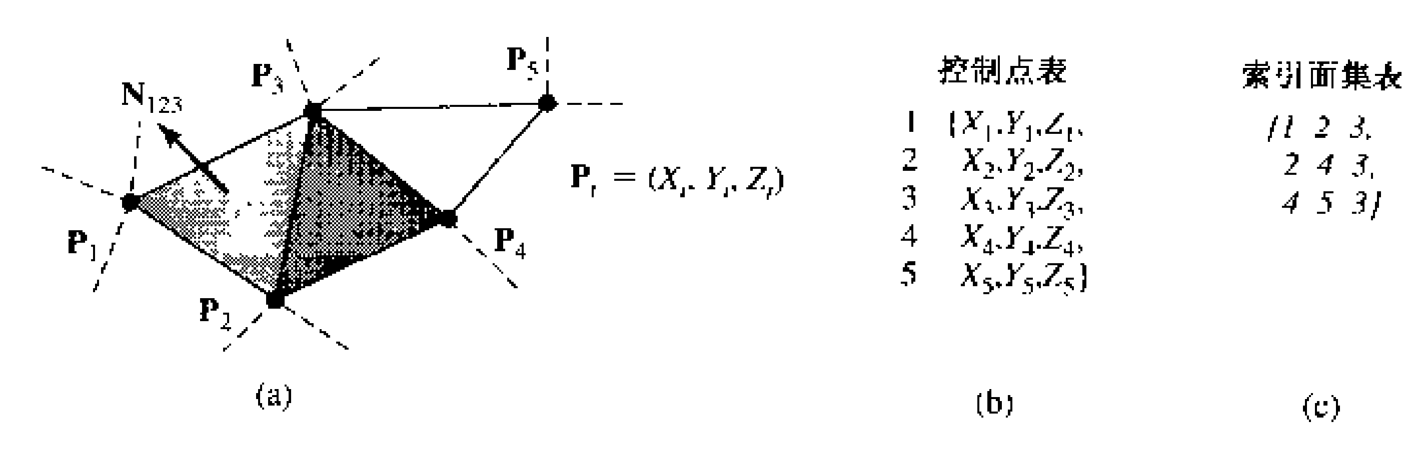


图 5.8 用于物体形状描述的线框：(a)表示物体表面的三角形网格；(b)三角形网格表示为一个控制点表；(c)一组关联控制点表的三角形表

一个线框的控制点位于物体的表面。控制点的数目和位置是由物体的形状以及线框模型对物体形状描述的精度决定的。当控制点不能被独立地移动时，这个物体就是刚性的，而且不能改变它的形状。否则，物体是柔性的，可以改变它的形状。

5.3.2 运动模型

正如刚刚讲到的,一个物体可以是刚性的或柔性的。刚性物体的运动可以用一个平移向量 $\mathbf{T}=(T_x,T_y,T_z)^T$ 和一个旋转矩阵[**R**]来描述。让我们首先用运动参数 **T** 和[**R**]描述物体上的一个点在三维空间是如何运动的。平移矢量 **T** 分别用在坐标轴 X,Y,Z 方向上的 T_x,T_y,T_z 描述一个点从 **X** 到 **X**′的位移:

$$\mathbf{X}'=\mathbf{X}+\mathbf{T} \tag{5.3.1}$$

如果物体被平移,那么公式(5.3.1)对物体上所有的点都是成立的。

如果一个物体绕着三维空间的原点转动,那么我们用旋转矩阵[**R**]来描述它的各个点的运动:

$$[\mathbf{R}]=[\mathbf{R}_z]\cdot[\mathbf{R}_y]\cdot[\mathbf{R}_x] \tag{5.3.2}$$

旋转矩阵按照这个顺序绕着 X,Y 和 Z 轴旋转一个点。它可以由只绕着一个轴旋转的几个旋转矩阵计算出来。它们各自的旋转矩阵是:

$$[\mathbf{R}_x]=\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\theta_x & -\sin\theta_x\\ 0 & \sin\theta_x & \cos\theta_x\end{bmatrix} \tag{5.3.3}$$

$$[\mathbf{R}_y]=\begin{bmatrix}\cos\theta_y & 0 & \sin\theta_y\\ 0 & 1 & 0\\ -\sin\theta_y & 0 & \cos\theta_y\end{bmatrix} \tag{5.3.4}$$

和

$$[\mathbf{R}_z]=\begin{bmatrix}\cos\theta_z & -\sin\theta_z & 0\\ \sin\theta_z & \cos\theta_z & 0\\ 0 & 0 & 1\end{bmatrix} \tag{5.3.5}$$

其中 θ_x,θ_y 和 θ_z 是关于每一个轴的旋转角度。最后,我们把[**R**]表示为旋转角度 θ_x,θ_y 和 θ_z 的函数:

$$[\mathbf{R}]=\begin{bmatrix}\cos\theta_y\cos\theta_z & \sin\theta_x\sin\theta_y\cos\theta_z-\cos\theta_x\sin\theta_z & \cos\theta_x\sin\theta_y\cos\theta_z+\sin\theta_x\sin\theta_z\\ \cos\theta_y\sin\theta_z & \sin\theta_x\sin\theta_y\sin\theta_z+\cos\theta_x\cos\theta_z & \cos\theta_x\sin\theta_y\sin\theta_z-\sin\theta_x\cos\theta_z\\ -\sin\theta_y & \sin\theta_x\cos\theta_y & \cos\theta_x\cos\theta_y\end{bmatrix} \tag{5.3.6}$$

这个旋转矩阵[**R**]是一个标准正交矩阵;即它满足:

$$[\mathbf{R}]^T=[\mathbf{R}]^{-1} \tag{5.3.7}$$

$$\det[\mathbf{R}]=\pm 1 \tag{5.3.8}$$

现在,我们可以把物体表面上的点 **X** 从 **X** 到 **X**′的运动表示为:

$$\mathbf{X}'=[\mathbf{R}]\cdot\mathbf{X}+\mathbf{T} \tag{5.3.9}$$

对于许多运动估计算法，按照公式(5.3.2)，非线性旋转矩阵必须关于旋转角度进行线性化处理。假设旋转角度很小，使得 $\cos(\alpha)\approx 1$，$\sin(\alpha)\approx\alpha$ 成立，则公式(5.3.6)简化为：

$$[\mathbf{R}]\approx[\mathbf{R}']=\begin{bmatrix}1 & -\theta_z & \theta_y\\ \theta_z & 1 & -\theta_x\\ -\theta_y & \theta_x & 1\end{bmatrix}\tag{5.3.10}$$

公式(5.3.9)使物体表面上的点 **X** 绕着世界坐标系统的中心旋转。在物体远离这个坐标系统的中心、而且物体只绕着它自身的中心旋转的情况下，需要一个大的平移矢量来表示它的运动。这种情况可以用下述方法避免：对每个物体定义一个局部坐标系统，并且定义物体关于它自身的中心 $\mathbf{C}=(C_x, C_y, C_z)^T$ 的旋转和平移，这个中心也可以是局部坐标系统的中心(见图5.9)，即：

$$\mathbf{X}'=[\mathbf{R}]\cdot(\mathbf{X}-\mathbf{C})+\mathbf{C}+\mathbf{T}\tag{5.3.11}$$

设 $\mathbf{C}\neq 0$，则公式(5.3.9)和(5.3.11)中的旋转矩阵是相同的，然而对于同样的物理运动平移矢量是不同的。

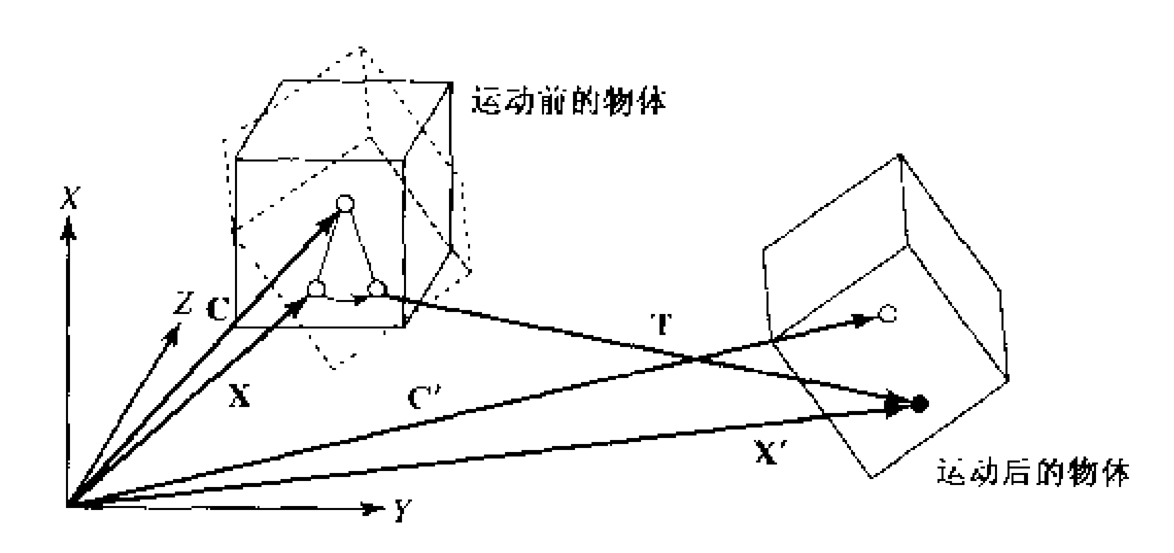


图 5.9 表示点 **X** 的运动的世界坐标系统

如果一个物体的所有点以相同的三维运动参数运动，则它是刚性的。正如前面所说明的，刚性物体的运动可以用参数 $A^{(m)}=(T_x^{(m)}, T_y^{(m)}, T_z^{(m)}, \theta_x^{(m)}, \theta_y^{(m)}, \theta_z^{(m)})$ 来描述。当一个刚性物体以参数集 $A^{(m)}$ 运动时，不改变它的形状。

不是所有真实的物体都是刚性的。一种描述柔性物体的方法是将这个物体分解成两个以上的刚体组元。按照公式(5.3.11)，每个组元都有它自己的运动参数。由于每个组元都是由它的控制点定义的，这些组元是由将不同组元中的控制点相互连接起来的三角形建立联系的。通过这些三角形，组元被柔性地连接起来。图 5.10 示出了一个具有物体"背景"和"克莱尔"(Claire)的场景。模型物体"克莱尔"包括两个组元"头"和"肩"。描述柔性物体的另一种方法是将小的局部运动叠加到刚性运动参数 $A^{(m)}$ 上。如果有一个带旗子的航行着的船，我们可以用一系列刚性参数 $A^{(m)}$ 描述整个船的运动，并且用局部运动对风中旗子的运动建模。由于柔性物体会因局部运动改变它们的形状，因此这种柔性应该用形状参数还是运动参数来描述不是明显的。

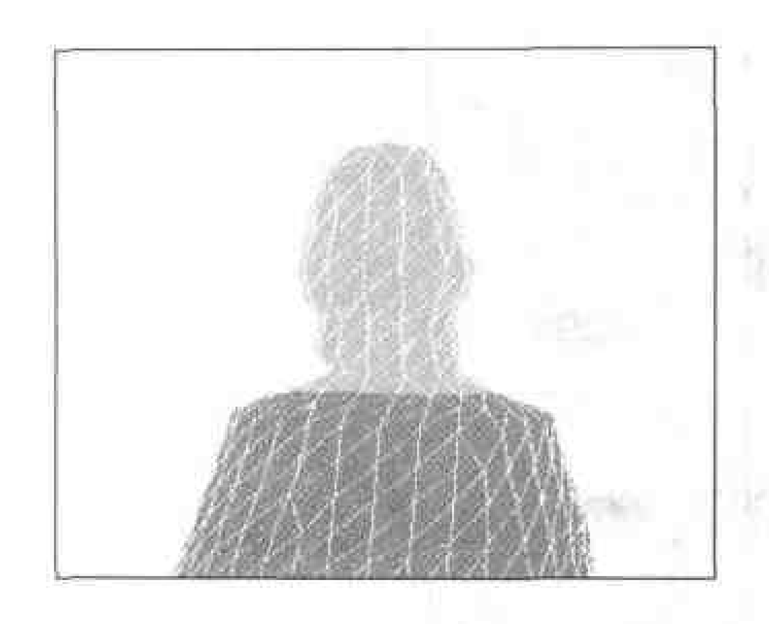



图 5.10 一个包括两个组元的模型物体。形状用线框表示，表面纹理是从图像中提取出来的

5.4 场景模型

上面已经讨论了如何对照明光源、物体和摄像机进行模型化，我们现在准备讨论成像场景的模型化。场景模型描述具有照明光源、物体和摄像机的世界。根据用于不同组元的模型，我们可得到不同的场景模型。图 5.11 画出了一个复杂的模型场景，它是用三维场景模型生成的。它是用一个假设是透视投影的摄像机模型得到的。这个模型可以用来描述物体的运动和物体的遮挡以及投射阴影。根据摄像机模型，每一条虚线描述一条观测线。物体表面上的一点被投影到图像平面上观测线与图像平面的交点。图像平面上的投影随着物体的运动面变化。物体 1 和 3 遮挡了物体 2 的一部分。所有的物体遮挡了图像背景的一部分，图像背景定义为在场景中没有出现运动物体时我们会得到的图像。（由于摄像机模型的透视投影，物体 3 延伸到未被物体 2 覆盖的图像区。图 5.11 中的场景只被散射光照射，因此我们看不到任何阴影。）如果物体是运动的，我们把它区分成四个图像区域：静止背景、运动物体、未覆盖的背景（或区域）以及新近覆盖的背景（或区域）。新近覆盖的区域也称为遮挡的区域或背景。

图 5.12 画出了具有一个运动物体的两个图像帧。比较帧 k 和 $k+1$，我们可以区分变化的和未变化的区域。未变化区域表示两幅图像中静止的背景。运动物体是图像 k 和 $k+1$ 中变化区域的一部分。在帧 k，变化区域定义为运动物体区以及由于物体的运动在帧 $k+1$ 将被覆盖的背景。在帧 $k+1$，变化的区域定义为运动物体区以及在帧 k 看不见的未覆盖的背景。

在图 5.13 中，我们看到一个具有分层的二维物体的场景。摄像机模型采用正交投影而不是透视投影。因此深度对场景的图像没有影响。我们可以描述从摄像机观测时，物体是如何一层摞一层的。这个场景模型不能描述三维运动的效果，因为我们没有明确地使用第三维空间，因而通常称为 2.5-D 场景模型。MPEG-4 视频编码标准支持视频物体分层，因此也支持这个场景模型（将在第 13 章讨论 MPEG-4）。

图 5.13 中的物体 1,2,3 处于与图 5.11 中相同的三维位置。由于使用了平行投影，光线现在都是平行的。因此，在图像平面中物体 3 不再延伸到物体 2 之外。

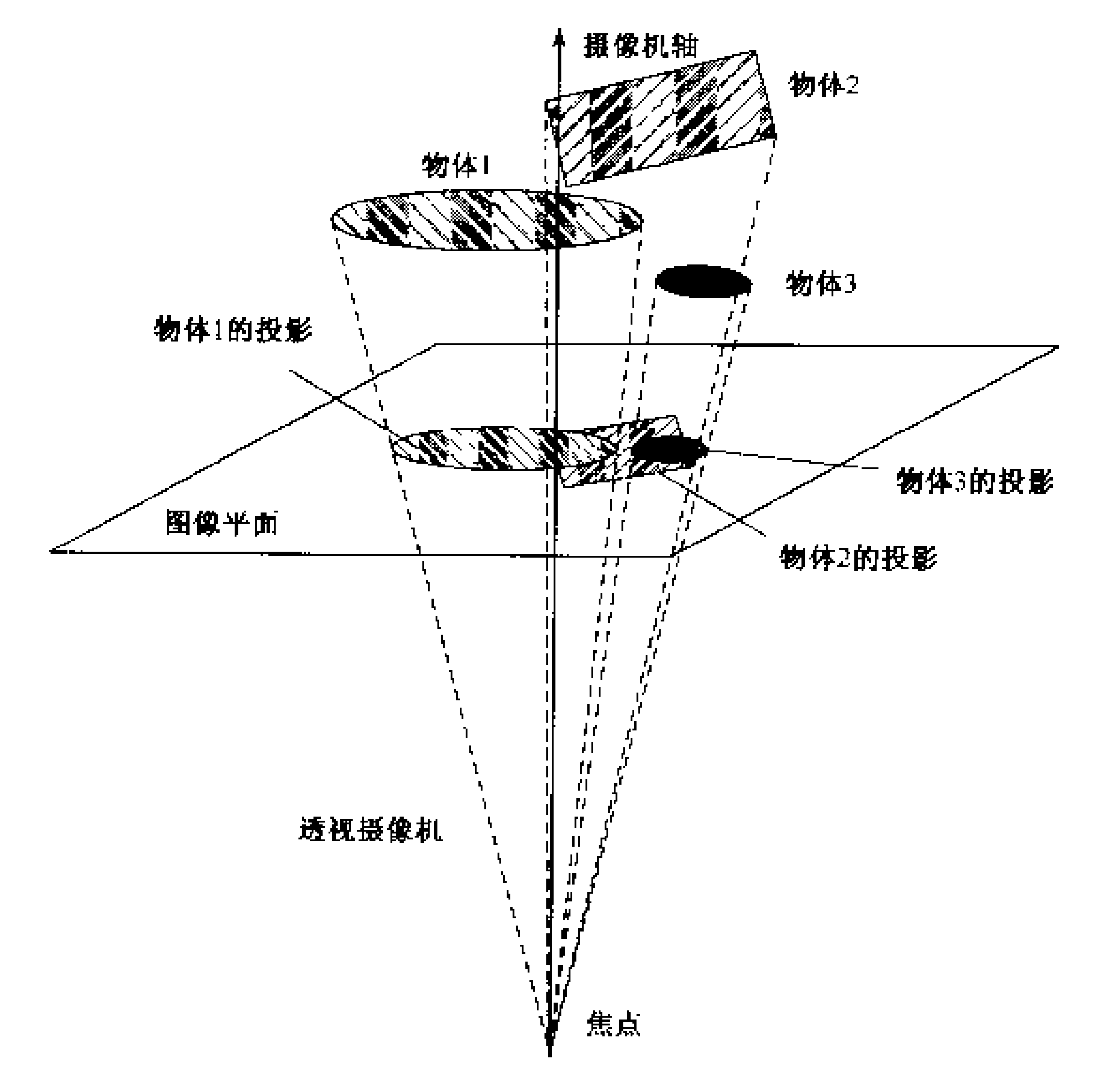


图 5.11 一个复杂的具有几个物体、摄像机和漫反射照明(未画)的三维场景

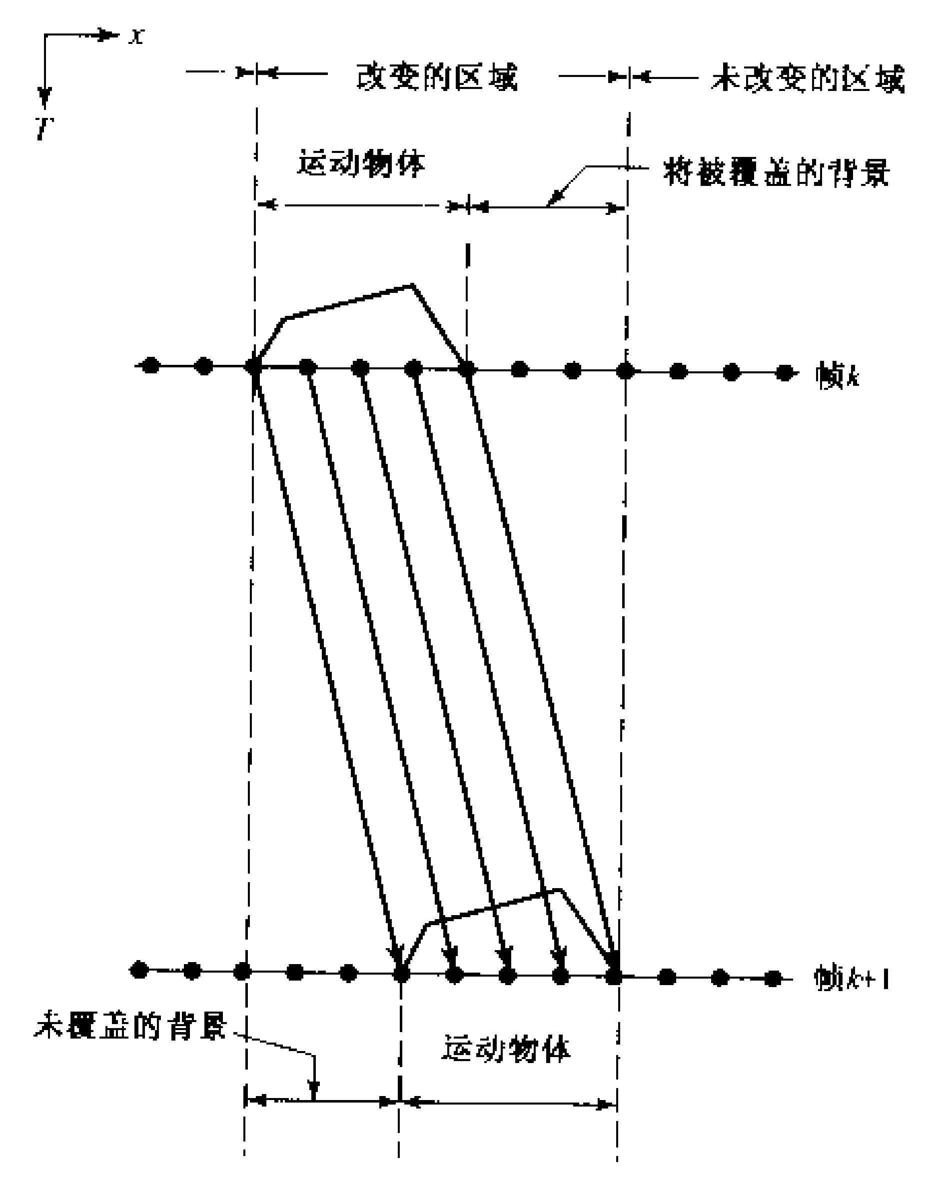


图 5.12 将变化的区域分成运动物体、未覆盖背景和即将被覆盖的背景

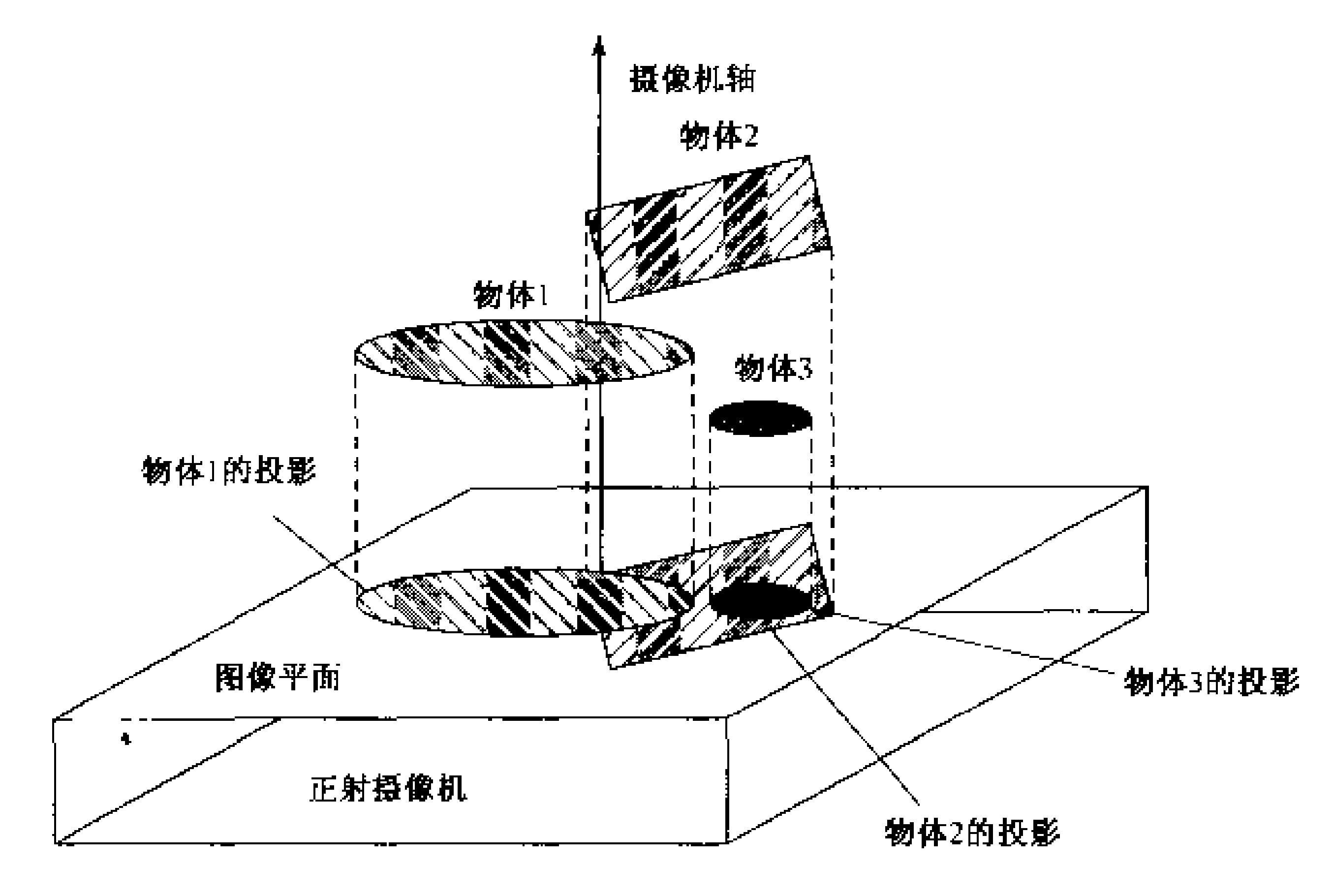


图 5.13 一个具有几个物体、一个摄像机以及环境照明(未画)的 2.5-D 场景

图 5.14 示出了对图像处理很有用的最简单的场景模型。它假设所有的物体都是平坦的,并且都在相同的图像平面上。二维物体被限制在一个二维平面上运动。不支持物体分层的概念。这个简单的二维场景模型被用于视频编码标准 H.261,H.263, MPEG-1 和 MPEG-2(将在第 13 章讨论)。

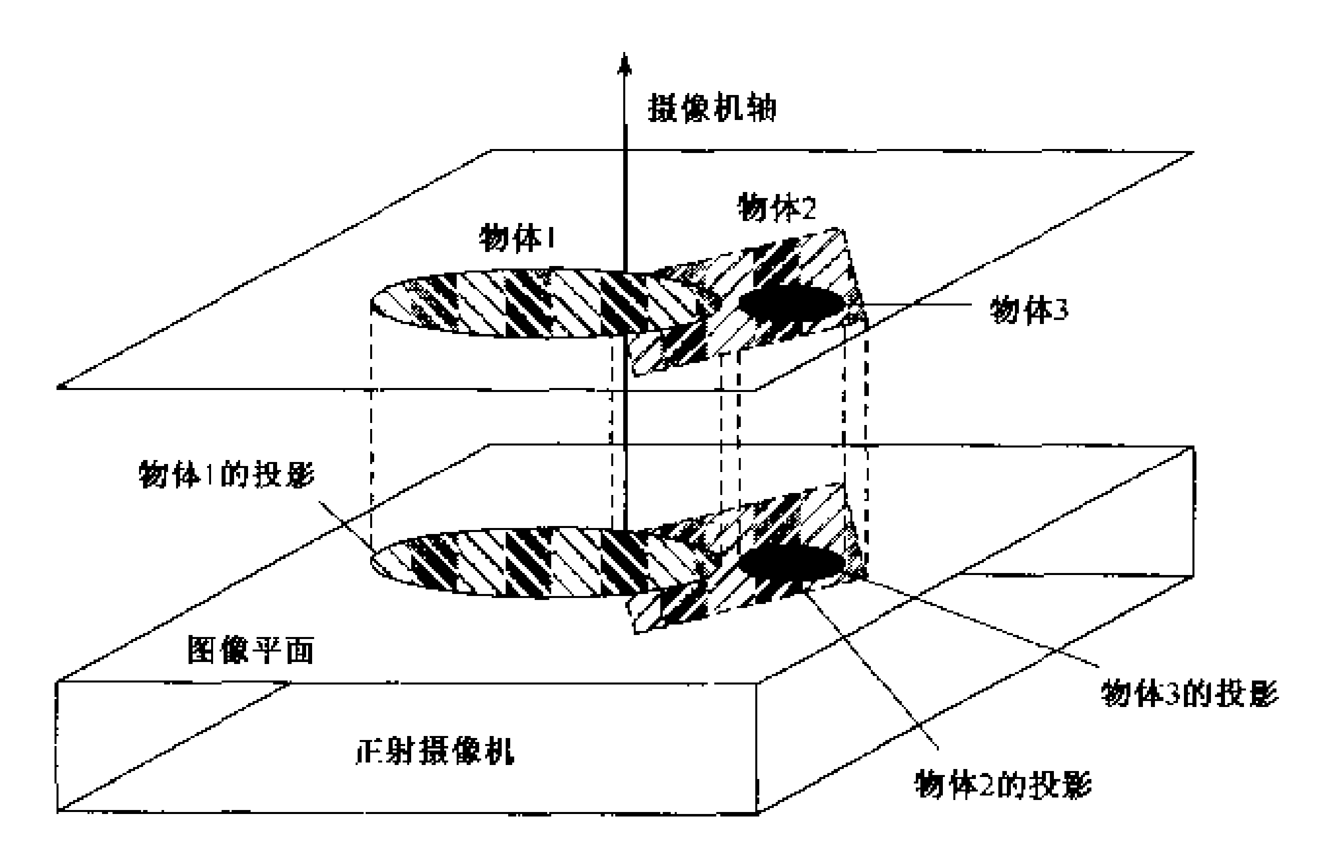


图 5.14 一个具有几个物体、一个摄像机和环境照明(未画)的二维场景

专用场景模型的选取会限制照明、摄像机和物体模型的选取。如果我们选择了一个二维场景模型,那么再选一个能够精确描述三维形状的物体模型就没有什么意义了。然而对于一些应用场合,选取一个三维场景模型的同时再选取一个只允许二维形状的物体模型可能是很有用的。

5.5 二维运动模型

5.5.1 定义和符号

当摄像机或场景中的物体运动时，同一个三维物体点的图像也将变化。图 5.15 画出了一个运动的物体在图像平面上的投影，以及二维和三维运动之间的关系。当一个物体点在时刻 t_1 从 $\mathbf{X}=[X,Y,Z]^T$ 运动到时刻 $t_2=t_1+d_t$ 的 $\mathbf{X}'=[X',Y',Z']^T=[X+D_X,Y+D_Y,Z+D_Z]^T$ 时，它的投影图像从 $\mathbf{x}=[x,y]^T$ 变为 $\mathbf{x}'=[x',y']^T=[x+d_x,y+d_y]^T$。我们称三维位移 $\mathbf{D}(\mathbf{X};t_1,t_2)=\mathbf{X}'-\mathbf{X}=[D_X,D_Y,D_Z]^T$ 为 $\mathbf{X}$ 处的三维运动矢量，称二维位移 $\mathbf{d}(\mathbf{x};t_1,t_2)=\mathbf{x}'-\mathbf{x}=[d_x,d_y]^T$ 为 $\mathbf{x}$ 处的二维运动矢量。注意，一般运动矢量(MV)是和位置有关的。作为在时刻 t_1 所有图像位置 $\mathbf{x}$ 的函数，$\mathbf{d}(\mathbf{x};t_1,t_2)$ 表示从 t_1 到 t_2 的一个二维运动场。当清楚所隐含的 t_1 和 t_2 时，可将其简写为 $\mathbf{d}(\mathbf{x})$。

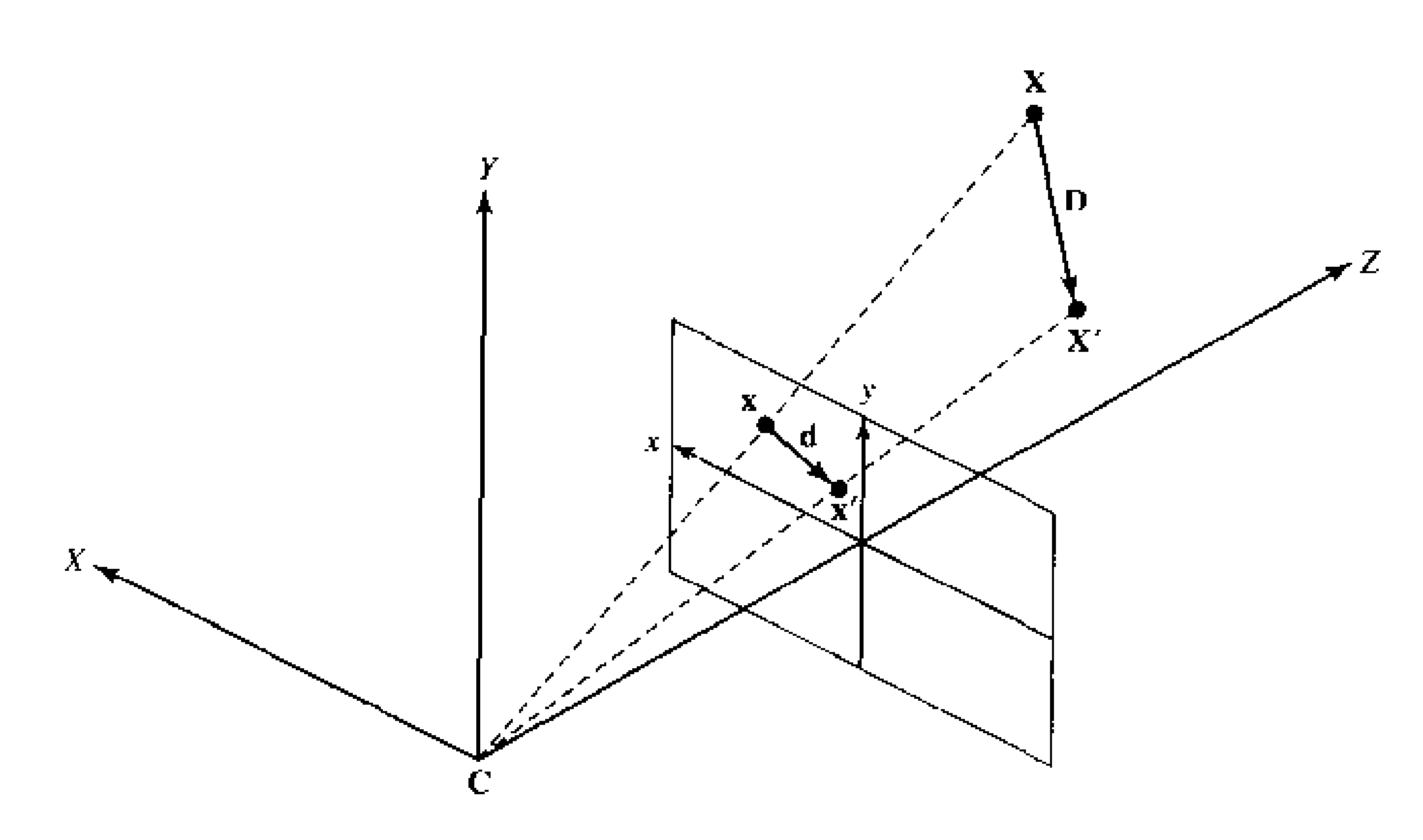


图 5.15 一个运动物体的投影：三维和二维运动矢量之间的关系

替换描述相应点之间的位移，有时更方便的是，对于 t_1 时刻的每个点 $\mathbf{x}$，指定它在 t_2 时刻相应的位置 $\mathbf{w}(\mathbf{x};t_1,t_2)=\mathbf{x}'$。我们称 $\mathbf{w}(\mathbf{x};t_1,t_2)=\mathbf{x}'$ 或者简化形式 $\mathbf{w}(\mathbf{x})$ 为映射函数。显然，它与运动场有惟一的关系：$\mathbf{w}(\mathbf{x})=\mathbf{x}+\mathbf{d}(\mathbf{x})$。

在本书中，我们只涉及具有有限和离散图像域的、被一个截断点阵 Λ 描述的数字视频信号。符号 $\mathbf{x}=[x,y]^T\in\Lambda$ 表示一个像素索引。我们进一步假设，时间间隔 $d_t=t_2-t_1$ 可以等于时间的采样间隔(即帧间隔)，也可以等于这个间隔的整数倍。对于一个给定的时间间隔，运动场是一个以与像素次序相同的二维阵列排列的二维矢量的有限集。这个离散运动场通常用一个矢量图画出，如图 5.16 所示。图中每个箭头的方向和幅度表示位于箭头起点像素处的 MV 的方向和幅度。

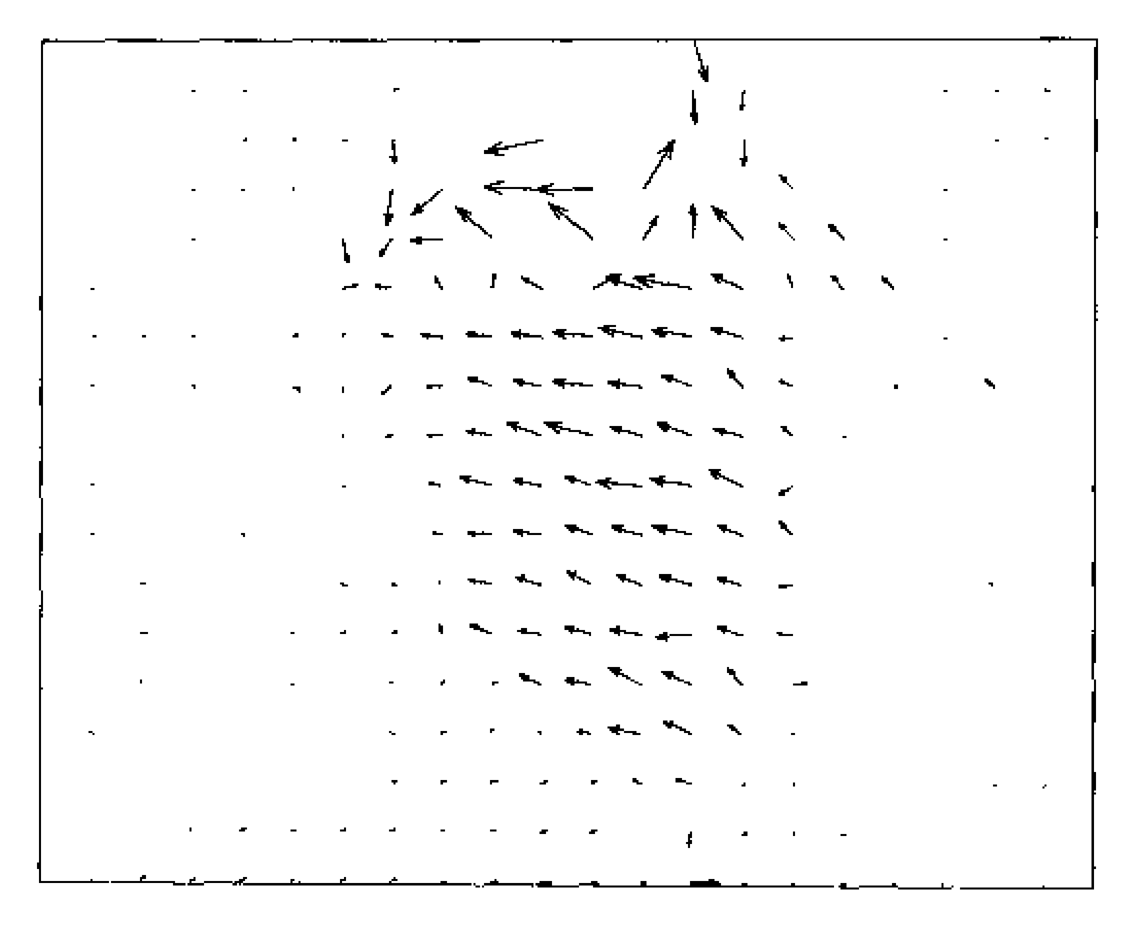


图5.16　一个典型的二维运动场

替换使用在给定时间间隔的实际的位移，我们可以用速度矢量来表征运动。这个速度矢量称为流矢量，定义为 $\mathbf{v}=\frac{\partial \mathbf{d}}{\partial t}=\left[\frac{\partial d_x}{\partial t},\frac{\partial d_x}{\partial t}\right]^T$。如果时间间隔足够小，那么在这个时间间隔内的运动就可以认为是常数，即 $\mathbf{v}=\mathbf{d}/d_t$。在处理数字视频时，我们通常认为 $d_t=1$，以便使流矢量与运动矢量同义。如定义运动场一样，我们可以定义整个图像域内的流场为 $\mathbf{v}(\mathbf{x};t_1,t_2)$，$\mathbf{x}\in\Lambda$。

在实际的视频序列中，运动矢量不总是定义在每个点上的。如5.4节所述，在一个场景中可能有重叠的物体，一个在另一个的后面。最普通的例子（见图5.12）是一个在静止的背景前运动的物体。未覆盖背景的运动矢量不定义。在任意给定时间，二维运动场的一个完整的定义应该包含一个掩模图像和一个运动场图像。例如，在图5.12中，如果运动是从帧 $k+1$ 到帧 k 定义的，那么在帧 $k+1$ 中未覆盖的背景中的像素应该标明是未覆盖的。

5.5.2　相应于典型的摄像机运动的二维运动模型

本节将描述相应于图5.4画出的一些摄像机运动的二维运动模型。这些模型可以由假设图5.2(a)所示的透视成像几何导出。即图像坐标$(x,\ y)$与世界坐标(X,Y,Z)具有公式(5.1.2)给出的透视映射关系。

为了导出由摄像机运动产生的二维运动，我们首先确定物体点关于新的摄像机坐标的新三维坐标，然后再用公式(5.1.2)确定它的图像位置。

在图像平面内的摄像机平移（跟和吊）　回忆跟和吊分别是指光学中心沿着 X 和 Y 轴的平移（见图5.4(a)）。令 T_x 和 T_y 表示实际的平移，那么在旧的摄像机坐标中任何一点(X,Y,Z)的三维位置将变为：

$$\begin{bmatrix}X'\\Y'\\Z'\end{bmatrix}=\begin{bmatrix}X\\Y\\Z\end{bmatrix}+\begin{bmatrix}T_x\\T_y\\0\end{bmatrix}\qquad(5.5.1)$$

应用公式(5.1.2),成像位置(x,y)变为:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} FT_x/Z \\ FT_y/Z \end{bmatrix}$$

因此像素(x, y)的平移取决于它所对应的三维点的 Z 坐标。然而,当成像物体深度的相对变化与物体到摄像机中心之间的平均距离 $\bar{Z}$ 相比很小时,我们可以用一个常矢量来近似这个运动场,即:

$$\begin{bmatrix} d_x(x,y) \\ d_y(x,y) \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

其中 $t_x = FT_x/\bar{Z}$、$t_y = FT_y/Z$。注意,沿着 Z 轴的平移(即推)将导致实际上更复杂的二维运动场。

摄像机摇和倾 摇和倾分别是指摄像机绕着 Y 和 X 轴旋转(见图5.4(b))。令 θ_y 和θ_x 表示相应的旋转角。摄像机的新坐标与旧坐标的关系是:

$$\mathbf{X}' = [\mathbf{R}_x][\mathbf{R}_y]\mathbf{X} \tag{5.5.2}$$

其中$[\mathbf{R}_x]$和$[\mathbf{R}_y]$如前面公式(5.3.3 及 5.3.4)所定义。当旋转角很小时,我们有:

$$[\mathbf{R}_x][\mathbf{R}_y] = \begin{bmatrix} 1 & 0 & \theta_y \\ 0 & 1 & -\theta_x \\ -\theta_y & \theta_x & 1 \end{bmatrix} \tag{5.5.3}$$

如果 $Y\theta_x \ll Z, X\theta_y \ll Z$, 那么 $Z' \approx Z$。用公式(5.1.2),我们得到:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \theta_y F \\ -\theta_x F \end{bmatrix} \tag{5.5.4}$$

或

$$\begin{bmatrix} d_x(x,y) \\ d_y(x,y) \end{bmatrix} = \begin{bmatrix} \theta_y F \\ -\theta_x F \end{bmatrix} \tag{5.5.5}$$

前面的结果表明当摇和倾的角度很小时,且当成像物体远离摄像机使得它在 X、Y 方向的延伸比它到摄像机的距离小得多时,物体的图像是一致移动的。

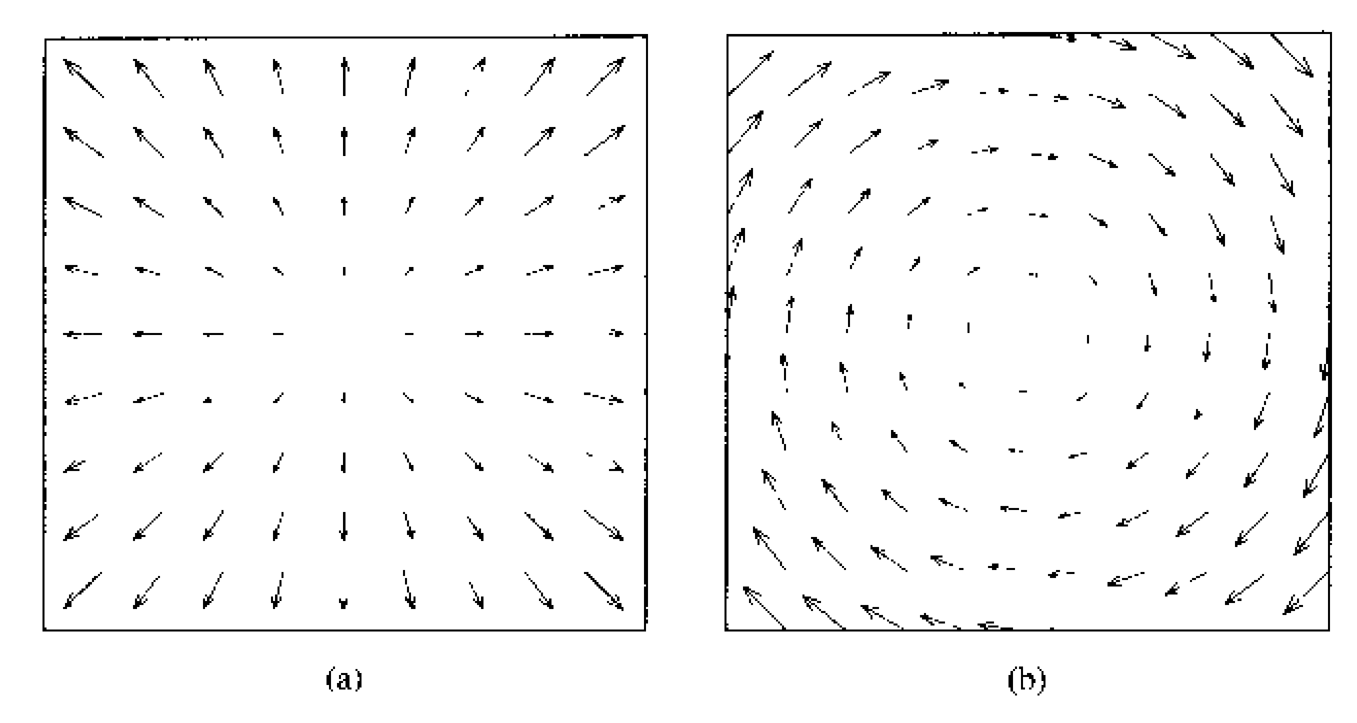
摄像机变焦 令 F 和 F' 表示变焦前和变焦后的焦距。用公式(5.1.2),容易证明:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \rho x \\ \rho y \end{bmatrix} \tag{5.5.6}$$

其中 $\rho = F'/F$ 是变焦系数。二维运动场是:

$$\begin{bmatrix} d_x(x,y) \\ d_y(x,y) \end{bmatrix} = \begin{bmatrix} (1-\rho)x \\ (1-\rho)y \end{bmatrix} \tag{5.5.7}$$

一个典型的对应于变焦的运动场示于图 5.17(a)中。



(a)　(b)

图 5.17　相应于(a)摄像机变焦;(b)摄像机旋转的运动场

摄像机滚　回忆滚是指绕 Z 轴旋转(见图 5.4(b))。因为这种旋转不改变 Z 坐标,所以透视投影后的净影响是图像坐标进行了旋转,即:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta_z & -\sin\theta_z \\ \sin\theta_z & \cos\theta_z \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \approx \begin{bmatrix} 1 & -\theta_z \\ \theta_z & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{5.5.8}$$

这里当 θ_z 很小时,近似是成立的。对应的运动场是:

$$\begin{bmatrix} d_x(x,y) \\ d_y(x,y) \end{bmatrix} = \begin{bmatrix} -\theta_z y \\ \theta_z x \end{bmatrix} \tag{5.5.9}$$

图 5.17(b)画出了由旋转引起的典型的运动场。

四参数模型　考虑一个顺序地进行平移、摇、倾、变焦和旋转的摄像机。运用前面给出的近似,同一个三维点的成像位置在摄像机运动前后的关系为:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \rho \begin{bmatrix} \cos\theta_z & -\sin\theta_z \\ \sin\theta_z & \cos\theta_z \end{bmatrix} \begin{bmatrix} x + \theta_y F + t_x \\ y - \theta_x F + t_y \end{bmatrix} \tag{5.5.10}$$

$$= \begin{bmatrix} c_1 & -c_2 \\ c_2 & c_1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c_3 \\ c_4 \end{bmatrix} \tag{5.5.11}$$

这个映射函数有四个参数,是仿射映射的一个特例,仿射映射一般有 6 个参数。这种特殊化的形式称为几何映射,它可以表征二维空间中放缩、旋转和平移的任意组合。如果摇、变焦和旋转的顺序改变了,这种形式将仍然是成立的,虽然会改变参数 c_k 和摄像机运动参数($\rho, \theta_x, \theta_y, \theta_z, t_x, t_y$)之间的关系。

5.5.3　相应于三维刚性运动的二维运动模型

前面的二维运动模型没有覆盖所有可能的摄像机三维运动。特别是,它们没有考虑摄像机在 Z 方向的移动,也没有考虑绕任意轴的旋转。三维空间中的任意一种摄像机的运动都可以分解为一个三维旋转和一个三维平移。类似地,一个成像的物体可能进行的是一个刚性运

动(旋转加平移),同时摄像机是静止的。一般,摄像机和物体都可能进行刚体运动,很难将摄像机和物体的运动分开。然而,整体的效果相当于假定一个项目(物体或摄像机)是静止的而另一个项目进行刚体运动。在本节中,我们考虑由这种任意的三维刚体运动引起的二维运动。不失一般性,我们将假设摄像机是静止的,而物体进行刚体运动。

一般情况 正如5.3节所述,任何物体点的三维位置在刚体运动前后的关系是:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \tag{5.5.12}$$

尽管旋转矩阵有9个参数,但用三个旋转角即可完全确定(见公式(5.3.6))。再加上3个平移参数,一共有6个参数。

将公式(5.1.2)代入公式(5.5.12),经过一些处理,我们可以得到运动前后图像坐标之间的关系:

$$x' = F\frac{(r_1x + r_2y + r_3F)Z + T_xF}{(r_7x + r_8y + r_9F)Z + T_zF}$$

$$y' = F\frac{(r_4x + r_5y + r_6F)Z + T_yF}{(r_7x + r_8y + r_9F)Z + T_zF} \tag{5.5.13}$$

注意,如果对平移参数 T_x, T_y, T_z 和深度 Z 进行同样因子的放缩,那么 (x, y) 与 (x', y') 之间的对应关系不会改变。这说明基于这种成像坐标的对应关系,参数 T_x, T_y, T_z 只相对于某个比例因子是惟一的。换句话说,只能确定平移矢量的方向而不能确定其长度。一般,当物体表面(即对应于每个 (x, y) 的 Z 值)是空间变化的时候,映射函数是逐点变化的,从而不能用少量的参数表征。

投影映射 当 Z 方向没有平移运动时,或者当成像物体具有一个平坦表面时[①],容易证明公式(5.5.13)可以简化为下面8个参数的投影映射[②]:

$$x' = \frac{a_0 + a_1x + a_2y}{1 + c_1x + c_2y}, \quad y' = \frac{b_0 + b_1x + b_2y}{1 + c_1x + c_2y} \tag{5.5.14}$$

在这个投影映射中有8个自由参数,它们依赖于5个运动参数(回忆三个平移参数只相对于某个比例是惟一的,因此它们中只有两个是自由的)和三个物体表面的平面参数。

在研究两个视频帧之间的运动或等效地研究两个视频帧的配准时,这个投影映射是一个重要的关系式。它精确地模型化了在摄像机或物体运动后同一物体的图像之间的关系,如果下述任何一个条件成立:(1)物体具有一个平坦表面(在这种情况下,摄像机或物体抑或两者都可进行任意的三维旋转或平移);或者(2)摄像机或物体沿成像轴(即Z轴)都没有平移[③](在这种情况下,物体的几何形状可以是任意的)。实际上,当成像场景不具有平坦表面时,我们可

① 这意味着对于某些常数a,b,c,物体上点的三维坐标满足 $aX + bY + cZ = 1$。

② 这个映射有时称为透视映射。这里,为了避免与透视成像系统中的三维和二维坐标间的关系混淆,我们用"投影"这个术语。

③ 一个在成像方向上没有平移的运动有时称为零视差运动。

以把成像区域划分成小的元素，使得对应于每个二维元素的三维小块是平坦的，并且对每个块运用投影模型。注意，投影映射包含作为特例的纯平移和仿射映射。

当成像几何可以用正交投影近似时(见公式(5.1.3))，容易证明一个进行三维刚体运动的平面小块可用一个仿射函数描述（见习题5.2)。

投影映射可以表征两个与透视成像有关的独特现象[6]，即(1)线性调频效果，它是指感觉的空间频率随着与摄像机距离的增大而增加；(2)会聚效果(也称为梯形失真效果)，它是指两束平行光随着距离的增加看起来越来越靠近，并且最终会聚为一个点。这些效果示于图5.18。

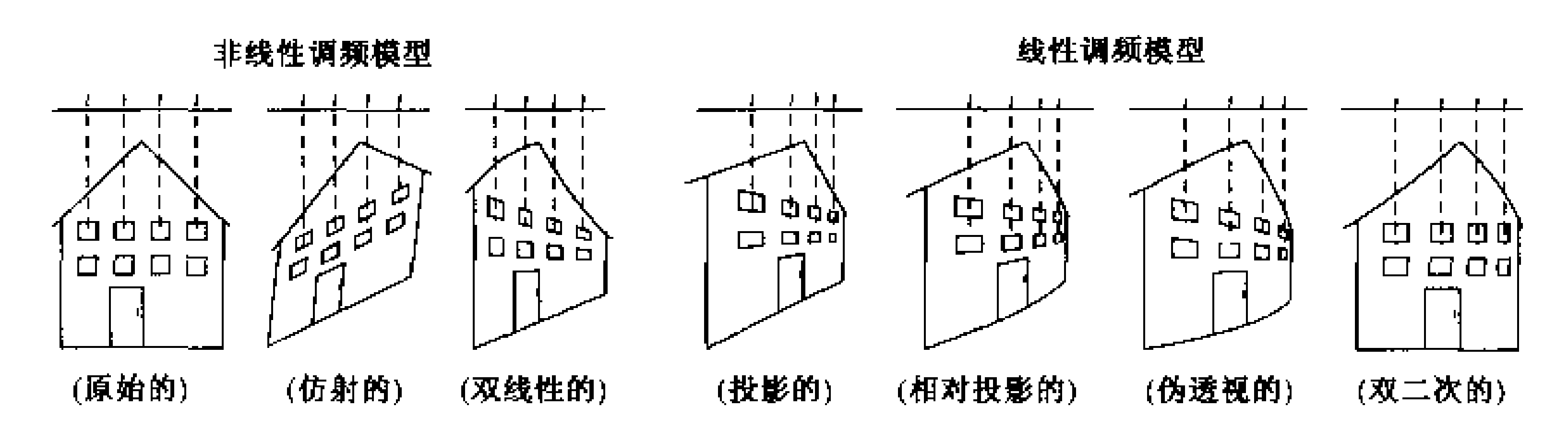


图5.18 不同映射函数效果的图解

投影映射的另一个重要特性是两个投影映射的级联是另一个投影映射[6]。这对于估计两个图像之间的投影映射是有意义的：我们可以用几步迭代来估计参数；只要每一步都引出一个投影映射，则总映射就保证是投影映射。

作用于一个正方形块的投影映射如图5.19(d)所示。透视函数可以在两个任意的四边形之间进行映射。代替用参数 a_l, b_l 和 c_l，我们可以用两个四边形之间四个角上的位移 $\mathbf{d}_k=(x'_k-x_k, y'_k-y_k)$，$k=1,2,3,4$，参数化一个投影映射。习题5.9考虑如何根据 $\mathbf{x}_k$ 和 $\mathbf{d}_k$ 确定投影映射参数。

相应于一个投影映射的运动场是：

$$d_x(x,y)=\frac{a_0+a_1x+a_2y}{1+c_1x+c_2y}-x,\quad d_y(x,y)=\frac{b_0+b_1x+b_2y}{1+c_1x+c_2y}-y. \tag{5.5.15}$$

应该指出的是运动场本身不能用投影函数描述。在[6]中，相应于一个运动场的映射用一个称为相对投影的投影映射作为近似。其效果示于图5.18。我们可以看出它能相对较好地近似于投影映射。

5.5.4 投影映射的近似

实际上，为了避开其有理形式引起的问题，投影映射通常由一个多项式映射来近似。在这一节中，我们讨论一些常用的近似。

仿射运动 仿射运动具有如下形式：

$$\begin{bmatrix} d_x(x,y) \\ d_y(x,y) \end{bmatrix}=\begin{bmatrix} a_0+a_1x+a_2y \\ b_0+b_1x+b_2y \end{bmatrix} \tag{5.5.16}$$

它有6个参数，$a_k, b_k, k=0,1,2$。正如我们已经指出的，通过投影，多数摄像机运动的二维运动可以用一个仿射函数描述。进一步，一个进行任意三维刚体运动的平面小块都可以用一个

正交投影下的仿射函数来描述。然而仿射模型不能获得投影映射的线性调频效果和会聚效果，如图 5.18 所示。

仿射运动可以形象化为因三角形角的移动由一个三角形变为另一个三角形。仿射参数完全是由这三个角的 MV 决定的。因此，一个仿射运动可以用这些 MV 参数化。在图 5.19(b)所示的例子中，我们只能指定三个角的位移，而第四个角的位移将由另外三个角的 MV 确定。

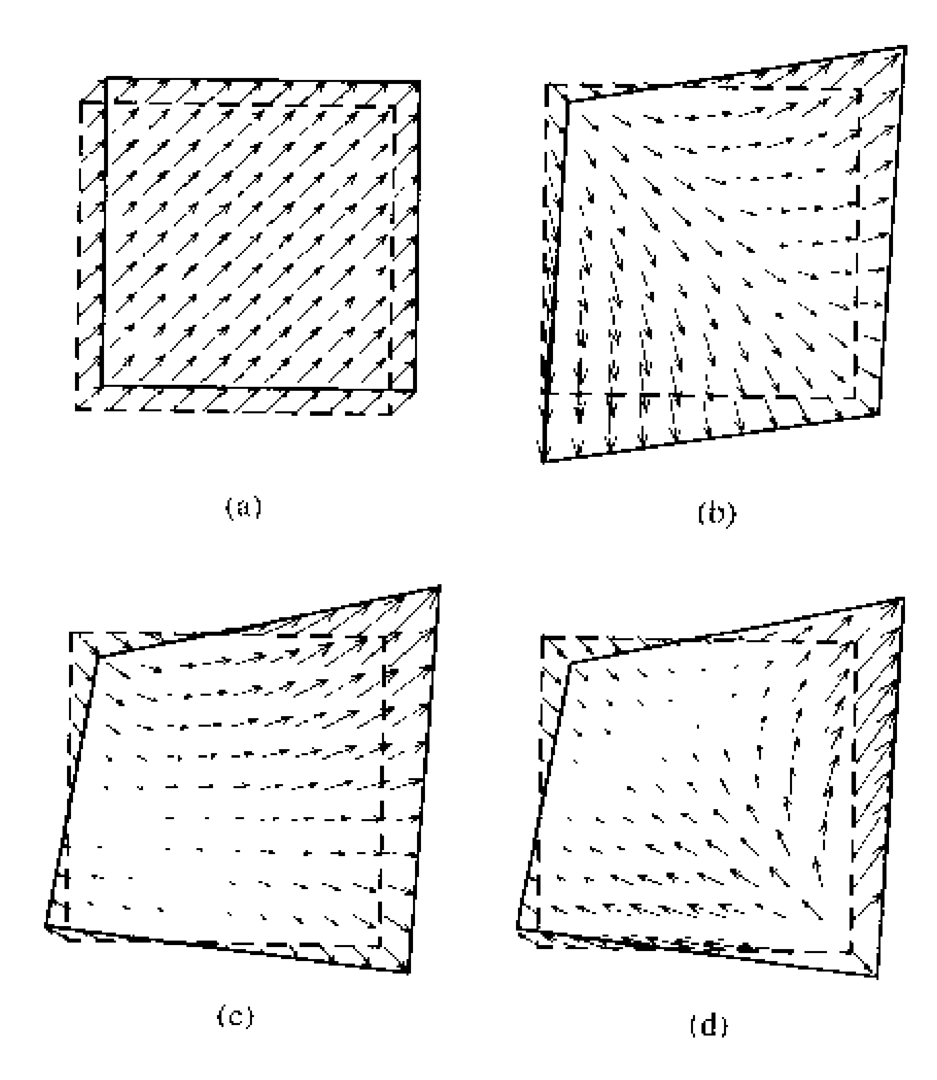


图 5.19　基本运动模型图示：(a)平移的；(b)仿射的；(c)双线性的；(d)投影的

双线性运动　双线性具有如下形式：

$$\begin{bmatrix} d_x(x,y) \\ d_y(x,y) \end{bmatrix} = \begin{bmatrix} a_0 + a_1 x + a_2 y + a_3 xy \\ b_0 + b_1 x + b_2 y + b_3 xy \end{bmatrix} \tag{5.5.17}$$

它有 8 个参数，$a_k, b_k, k=0,1,2,3$。从图 5.18 可以看到，双线性映射可以得到投影映射的会聚效果，但不能得到线性调频效果。

双线性运动可以形象化为一个正方形被扭曲成一个四边形，如图 5.19(c)所示。这 8 个参数由原始四边形的四个角的 MV 完全决定。注意，与投影映射不同，双线性函数不能在两个任意四边形之间进行映射，尽管这种映射只有 8 个自由度。这是因为除非原始线是水平的或垂直的，双线性函数会把一条直线映射成曲线。一般情况是双线性函数将一个四边形映射成一个曲边的四边形[15]。

其他多项式模型　通常任何运动函数都可用如下形式的多项式映射近似：

$$\begin{bmatrix} d_x(x,y) \\ d_y(x,y) \end{bmatrix} = \sum_{\substack{0 \le i,j \le N_1 \\ i+j \le N_2}} \begin{bmatrix} a_{i,j} \\ b_{i,j} \end{bmatrix} x^i y^j \tag{5.5.18}$$

平移、仿射以及双线性运动是这个模型的特例，当 $N_1 = N_2 = 0$ 时是平移，当 $N_1 = N_2 = 1$ 时是

仿射，当 $N_1=1, N_2=2$ 是双线性。相应于 $N_1=2, N_2=2$ 的情况是：

$$\begin{bmatrix} d_x(x,y) \\ d_y(x,y) \end{bmatrix} = \begin{bmatrix} a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 y^2 + a_5 xy \\ b_0 + b_1 x + b_2 y + b_3 x^2 + b_4 y^2 + b_5 xy \end{bmatrix} \tag{5.5.19}$$

这称为双二次映射，它有 12 个参数。如图 5.18 所示，双二次映射可以产生线性调频效果，但不能产生会聚效果。

在参考文献[6]中，Mann 和 Picard 对各种投影映射的多项式近似的优缺点进行了比较。他们发现，仿射函数尽管参数较少，但是较容易受到噪声的干扰，因为它缺乏完全跟踪实际图像运动所需的恰当的自由度。另一方面，双线性函数尽管具有恰当数量的参数，但不能得到线性调频效果。一个较好的 8 参数模型是如下的伪透视映射：

$$\begin{bmatrix} d_x(x,y) \\ d_y(x,y) \end{bmatrix} = \begin{bmatrix} a_0 + a_1 x + a_2 y + a_3 xy + b_3 x^2 \\ b_0 + b_1 x + b_2 y + b_3 xy + a_3 y^2 \end{bmatrix} \tag{5.5.20}$$

从图 5.18 看到，这个映射既能产生线性调频效果又能产生会聚效果，它是用低阶多项式对投影映射最好的近似。

5.6 小结

本章讨论了参数化模型，它使我们能够描述一个图形序列是如何产生的。根据应用场合，我们可以利用这些模型以适当的精度描述真实世界。

摄像机模型(5.1 节) 摄像机模型描述的是用真实摄像机将三维世界投影到图像平面。根据具体的应用场合，我们可以用具有不同复杂程度的摄像机模型。如果三维世界中的物体离摄像机很远，可以用具有正交投影的简单的摄像机模型。针孔摄像机模型使用透视投影，它使得我们能够随着物体到摄像机的距离变化，描述图像序列中物体尺寸的变化。

照明模型(5.2 节) 为了能看到物体，需要照明所观测的场景。描述物体表面光的照明和反射，通常需要复杂的模型。然而，我们假设了一个用于许多视频处理场合的最简单的照明模型：环境光和漫反射表面。在这种模型下，一个物体表面的反射系数不随物体的移动而改变。这个模型有时候称为恒定亮度假设(第 6 章和第 7 章给出的运动估值算法广泛地使用了这个假设)。注意，用这个模型不能描述阴影或光滑反射。

物体模型(5.3 节) 物体用形状、运动和纹理描述。一个物体的三维形状可以利用称为体素的单位正方体的体积描述；或者利用线框的表面描述。我们用一个三维平移矢量和一个由绕 3 个坐标轴的旋转角计算得到 3×3 旋转矩阵，来表示三维物体的运动。可以通过简化三维物体的形状和运动模型导出简单的模型。

场景模型(5.4 节) 场景模型描述运动物体和一个三维场景的摄像机是如何互相定位的。在视频编码中，我们通常使用二维场景模型，假设二维物体与摄像机的成像平面平行运动。稍微复杂一些的 2.5 维场景模型使我们能够处理物体遮挡。最后，三维场景模型可以逼真地描述真实世界。

二维运动模型(5.5 节) 在三维空间中物体或摄像机的运动导致二维运动。二维运动模型不仅取决于三维运动模型，同时也取决于照明和摄像机模型。最重要的二维运动模型是投

影映射,它适用于小的二维区域,当物体和摄像机进行刚体运动时,摄像机可用透视投影建模,而且适用恒定亮度假设。实际上,这种映射通常用仿射映射或双线性映射来近似。

5.7 习题

5.1 假设针孔摄像机的焦距为 $F = 9$ mm,光靶为 $1'' \times 1.33''$,图像分辨率为 352×288 像素,物体点到摄像机中心的距离为 $Z = 2$ m。确定作为三维空间位置(X, Y)的函数的点在图像上的投影。为了使它的成像点移动一个像素,那么该点需要在 Z 方向上移动多少? 如果我们假设摄像机模型是正交投影,那么答案又是什么?

5.2 试证明,在正交投影下,一个进行平移、旋转和放缩(由于摄像机变焦)的平坦小块的二维投影运动可以用一个仿射函数描述。

5.3 试证明,在透视投影下,一个进行刚体运动的平坦小块的二维投影运动可以用一个投影映射描述。

5.4 试证明当成像物体具有平坦表面时,公式(5.5.13)可以简化为公式(5.5.14)所给出的投影映射。也就是说,物体上点的三维坐标对于某些常数 a, b, c 满足 $aX + bY + cZ = 1$。

5.5 考虑一个三角形,它原始的角的位置在 $\mathbf{x}_k, k = 1,2,3$。假设每一个角移动了 $\mathbf{d}_k$。确定能够实现这种映射的仿射参数。

5.6 假设已知图像中点 $\mathbf{x}_k(K > 3)$的运动矢量 $\mathbf{d}_k$,现在想用仿射映射来近似整个运动。如何确定仿射参数?

(提示:用最小平方匹配法。)

5.7 对于两个四边形之间的双线性映射,重做习题 5.5。假设可以利用 $K = 4$ 个点处的位移。

5.8 对于双线性映射,重做习题 5.6。假设 $K > 4$。

5.9 对于两个四边形之间的投影映射,重做习题 5.5。假设可以利用 $K = 4$ 个点处的位移。

(提示:仍然可以建立一个参数为 a_k, b_k, c_k 的线性方程。在方程左边的常数将包括 $\mathbf{x}_k$ 和 $\mathbf{x}'_k = \mathbf{x}_k + \mathbf{d}_k$。)

5.10 对于投影映射,重做习题 5.6。假设 $K > 4$。

5.8 文献目录

[1] Bozdagi, G., A. M. Tekalp, and L. Onural. 3-D motion estimation and wireframe adaptation including photometric effects for model-based coding of facial image sequences. *IEEE Trans. Circuits Syst. for Video Technology* (June 1994), 4(3): 246–56.

[2] Faugeras, O. *Three-Dimensional Computer Vision—A Geometric Viewpoint*. Cambridge, MA: MIT Press, 1993.

[3] Foley, J., A. van Dam, S. K. Feiner, J. F. Hughes, and R. L. Phillips. *Introduction*

to Computer Graphics. Reading, MA: Addison-Wesley, 1994.

[4] Gilge, M., Motion estimation by scene adaptive block matching (SABM) and illumination correction. In *SPIE Image Processing Algorithms and Techniques* (Feb. 1990), 1244:355–66.

[5] Hötter, M., and R. Thoma. Image segmentation based on object oriented mapping parameter estimation. *Signal Processing* (Oct. 1988), 15(3):315–34.

[6] Mann, S., and R. W. Picard. Video orbits of the projective group: A simple approach to featureless estimation of parameters. *IEEE Trans. Image Process.* (Sept. 1997), 6:1281–95.

[7] Musmann, H. G., M. Hötter, and J. Ostermann. Object oriented analysis-synthesis coding of moving images. *Signal Processing: Image Commun.* (Oct. 1989), 1:119–38.

[8] Niem, W. Robust and fast modelling of 3-D natural objects from multiple views. *SPIE Image and Video Processing II* (1994), 2182:388–97.

[9] Ostermann, J. Segmentation of image areas changed due to object motion considering shadows. In Y. Wang, et al., eds., *Multimedia Communications and Video Coding*. New York: Plenum Press, 1996.

[10] Pollefeys, M., R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Int. Conf. Computer Vision (ICCV'98)*, 90–95.

[11] Rogers, D. F. *Procedural Elements for Computer Graphics*. Singapore: McGraw-Hill, 1985.

[12] Stauder, J. An illumination estimation method for 3-D-object-based analysis-synthesis coding. In *COST 211 European Workshop*, 4.5.1–6. University of Hannover, Germany, December 1993.

[13] Stauder, J. Estimation of point light source parameters for object-based coding. *Signal Processing: Image Communications* (1995), 46(7):355–79.

[14] Stauder, J., R. Mech, and J. Ostermann. Detection of moving cast shadows for object segmentation. *IEEE Transactions on Multimedia* (March 1999), 1(1):65–76.

[15] Wolberg, G. *Digital Image Warping*. Los Alamitos, CA: IEEE Computer Society Press, 1990.

[16] Yakimovsky, Y., and R. Cunningham. A system for extracting three-dimensional measurements from a stereo pair of TV cameras. *Computer Graphics and Image Processing* (April 1978), 7(2):195–210.

第6章 二维运动估计

运动估计是视频处理系统的一个重要的组成部分。在这一章中,我们讨论二维运动估计;在第7章,我们讨论真实的三维运动估计。正如将要看到的,二维运动估计时常是三维结构和运动估计所需要的一个预处理阶段。同样,二维运动估计本身也有着范围广泛的应用,包括视频压缩、采样率转换、滤波等。根据所期望的应用场合,为产生二维运动矢量,运动估计的方法可以是很不相同的。例如,对于计算机视觉应用场合,二维运动矢量用于推演三维结构和运动参数,在关键特征点上的一个稀疏的二维运动矢量集可能就足够了(为了利用它们,这些运动矢量必须在物理上是正确的)。另一方面,对于视频压缩场合,估计的运动矢量用来根据前一个已编码的参考帧产生当前编码帧的运动补偿预测。最终目的是使编码运动矢量和预测误差所用的总的比特数最小化。我们可以在运动估计的准确性与表示运动所用的比特数之间做出折中的选择。有时,即使估计的运动不是实际物理运动的精确表示,仍可以产生好的时间预测,并且从这一点考虑,被认为是一个好的预测。本章集中讨论以运动补偿处理(预测、滤波、内插等)为目的的运动估计算法的类型。关于这个课题的其他读物,读者可以参考 Musmann, Pirsch 和 Grallert[28]以及 Stiller 和 Konrad[38]的综述论文。关于计算机视觉应用中运动估计方法的一个很好的介绍,请参考 Aggarwal 和 Nandhahumar 的文章[1]。

所有运动估计算法都是基于图像亮度(更一般地,彩色)的时间变化的。实际上,基于亮度变化观察到的二维运动可能不同于真实的二维运动。为了更精确起见,把观测的或表观的二维运动矢量的速度称为光流。光流不仅可以由物体运动引起,而且可由摄像机运动或者照明条件的变化引起。本章从定义光流开始。然后导出光流方程,这个方程在图像梯度和流矢量之间施加了一个约束。这是许多运动估计算法所依据的一个基本等式。接着,我们给出二维运动估计的一般方法学。大家将会看到,运动估计问题通常转化为一个最优化问题,并且涉及三个关键部分:运动场参数化、最优化准则的公式化和寻找最优化参数。最后,我们给出已经开发出来的基于不同的运动场参数化和不同估计准则的运动估计算法。除非特别说明,“运动”一词在本章指的是二维运动。

6.1 光流

6.1.1 二维运动和光流

人眼是通过在不同的时刻认出相对应的一些点来感觉运动的。这种对应性通常是由假定一个点的彩色或亮度在运动以后不改变来确定的。有趣的是要注意,在某些情况下,观测到的二维运动可能不同于实际投影的二维运动。图6.1给出了两种特殊的情况。在第一个例子中,一个具有均匀平坦表面的球在恒定的环境光下转动。因为球上的每一点都反射同样的彩色,所以人眼观察不到在成像球的彩色图案中的任何变化,因此认为球是静止的。在第二个例子中,球是静止的,但是被一个绕着球转动的点光源照明。光源的运动引起球上反射光点的运

动,因而使人眼认为球在运动。观测的或表观的二维运动在计算机视觉文献中被称为光流。前面的例子说明光流可能不等同于真实的二维运动。当只能利用图像的彩色信息时,所能够得到的最精确估计就是光流。然而,在本章剩下的部分中,我们将使用术语"二维运动"或者简称"运动"来描述光流。读者一定要记住它有时可能不同于真实的二维运动。

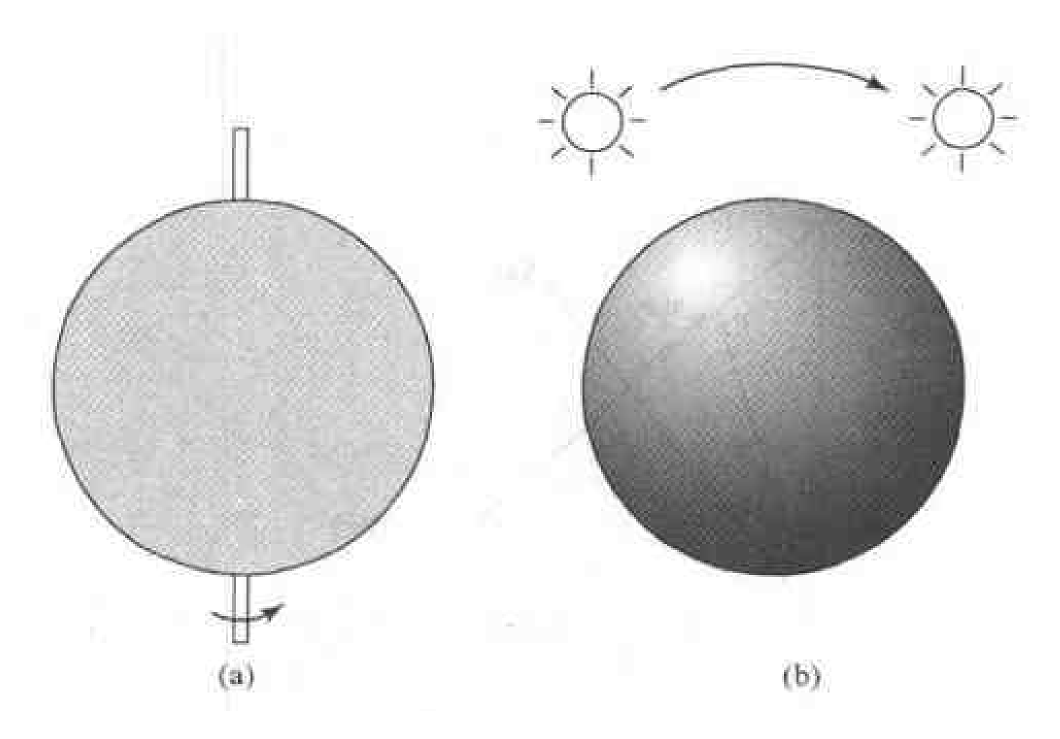


图6.1　光流并不总是与真实运动场相同。在图(a)中,球体在恒定环境照明下转动,但是观测的图像没有变化。在图(b)中,点光源绕着静止的球转动,引起球上的亮点旋转

6.1.2　光流方程和运动估计的多义性

考虑一个视频序列,它的亮度变化用$\psi(x,y,t)$表示[①]。假定在时刻t的一个成像点(x,y)在时刻$t+d_t$移动到点$(x+d_x,y+d_y)$。在5.2.3节(公式(5.2.11))介绍的恒定亮度假设下,同一个物体点在不同时刻的图像具有相同的亮度值。因此:

$$\psi(x+d_x,y+d_y,t+d_t)=\psi(x,y,t) \tag{6.1.1}$$

应用泰勒展开公式,当d_x,d_y,d_t很小时,我们有:

$$\psi(x+d_x,y+d_y,t+d_t)=\psi(x,y,t)+\frac{\partial\psi}{\partial x}d_x+\frac{\partial\psi}{\partial y}d_y+\frac{\partial\psi}{\partial t}d_t \tag{6.1.2}$$

联合公式(6.1.1)和(6.1.2)得到:

$$\frac{\partial\psi}{\partial x}d_x+\frac{\partial\psi}{\partial y}d_y+\frac{\partial\psi}{\partial t}d_t=0 \tag{6.1.3}$$

公式(6.1.3)是用运动矢量(d_x,d_y)写出的。两边同时除以d_t得到:

$$\frac{\partial\psi}{\partial x}v_x+\frac{\partial\psi}{\partial y}v_y+\frac{\partial\psi}{\partial t}=0\ 或\ \nabla\psi^T\mathbf{v}+\frac{\partial\psi}{\partial t}=0 \tag{6.1.4}$$

其中(v_x,v_y)表示速度矢量(也称为流矢量),$\nabla\psi=\left[\frac{\partial\psi}{\partial x},\frac{\partial\psi}{\partial y}\right]^T$是$\psi(x,y,t)$的空间梯度矢量。

① 在本书中,我们只考虑基于亮度强度信息的运动估计,虽然同样的方法也可应用于全彩色信息。

公式(6.1.4)通常称为光流方程,我们已经假定 d_t 很小,使得 $v_x = d_x/d_t, v_y = d_y/d_t$[①]。这个公式成立的条件与5.2.3节中所讨论的恒定亮度假设的条件相同。

如图6.2所示,在任何点 $\mathbf{x}$ 的流矢量 $\mathbf{v}$ 都可以分解为两个正交的分量:



$$\mathbf{v} = v_n \mathbf{e}_n + v_t \mathbf{e}_t \tag{6.1.5}$$

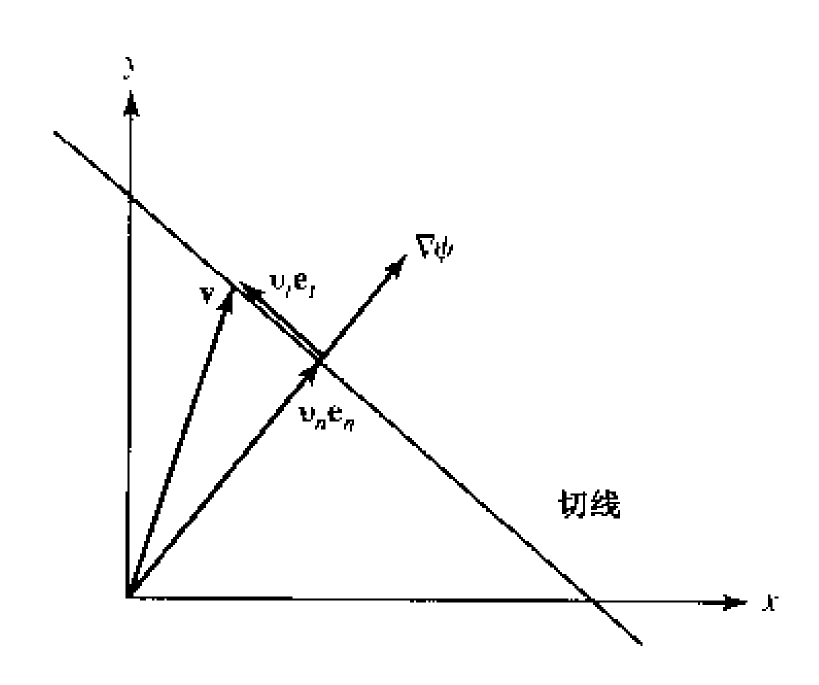


图6.2 把运动 $\mathbf{v}$ 分解成法线($v_n\mathbf{e}_n$)和切线($v_t\mathbf{e}_t$)分量。给定 $\nabla\psi$ 和$\frac{\partial\psi}{\partial t}$,在切线上的任何运动矢量都满足光流方程

这里 $\mathbf{e}_n$ 是图像梯度 $\nabla\psi$ 的方向矢量,称为法向。$\mathbf{e}_t$ 垂直于 $\mathbf{e}_n$,称为切向。公式(6.1.4)的光流方程可以写成:

$$v_n \| \nabla\psi \| + \frac{\partial\psi}{\partial t} = 0 \tag{6.1.6}$$

其中 $\| \nabla\psi \|$ 是梯度矢量的幅度。由公式(6.1.4)或者(6.1.6)推出的三个结果是:

1. 在任何像素 $\mathbf{x}$,我们不能单凭 $\nabla\psi$ 和 $\partial\psi/\partial t$ 确定速度矢量 $\mathbf{v}$。对于两个未知量(v_x 和 v_y,或者 v_n 和 v_t)只有一个方程。实际上,未确定的分量是 v_t。为了解出两个未知量,我们必须施加附加条件。最通常的约束是流矢量在空间平滑变化,使我们能利用 $\mathbf{x}$ 周围一个小的邻域的亮度变化去估计 $\mathbf{x}$ 处的运动。

2. 给定 $\nabla\psi$ 和 $\partial\psi/\partial t$,速度矢量在法向的投影是固定的,即 $v_n = -(\partial\psi/\partial t)/\| \nabla\psi \|$,但是在切向的投影 v_t 是不确定的。任何 v_t 的值都将满足光流方程。在图6.2中,这意味着任何切线上的点都将满足光流方程。这种在估计运动矢量时出现的多义性称为孔径问题,"孔径"一词在这里指的是应用恒定亮度假设的一个小窗口。只有当孔径至少包括两个不同的梯度方向时,才能做出惟一的运动估计,正如图6.3所示。

3. 在恒定亮度区域,即 $\| \nabla\psi \| = 0$,流矢量是不确定的。这是因为当表面具有平坦图形时,不会感觉到有亮度变化。运动估计仅在有亮度变化的区域是可靠的;换句话说,仅在有边界的或是非平坦的纹理区才是可靠的。

---

① 另一种导出光流方程的方法是把恒定亮度假设表示成 $d\psi(x,y,t)/dt = 0$。用偏微分展开 $d\psi(x,y,t)/dt$ 将得到同样的方程。

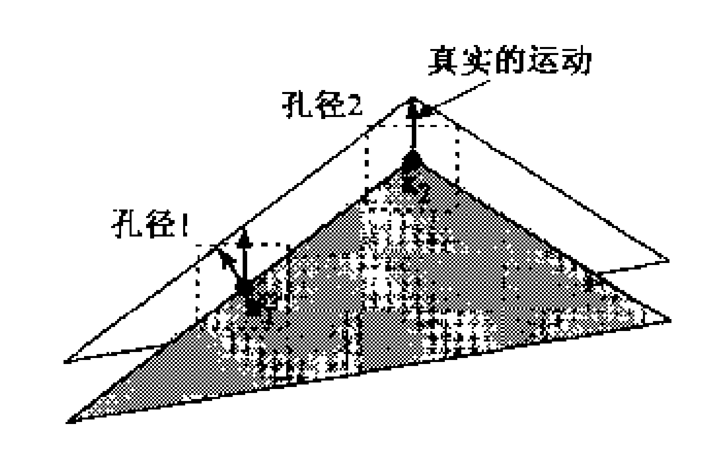


图 6.3 运动估计中的孔径问题：在 $\mathbf{x}_1$ 处估计运动使用孔径 1，但是不可能确定运动是向上的还是垂直于边缘的，因为在这个孔径中仅有一个空间梯度方向。另一方面，可以准确估计 $\mathbf{x}_2$ 处的运动，因为在孔径 2 中，图像有两个不同方向的梯度

上述观点与在 2.3.2 节中讨论的空间和时间频率间的关系是一致的。在那里，我们指出当空间频率为零时，或者当运动方向垂直于空间频率时，则运动物体的时间频率为零。当时间频率为零时，不会观察到成像图形的变化，因此，运动是不确定的。

在后面各节中将会看到，在所有运动估计算法中，光流方程（或者等效地，恒定亮度假设）起着关键的作用。

6.2 一般方法

本章考虑两个给定的帧 $\psi(x,y,t_1)$ 和 $\psi(x,y,t_2)$ 之间的运动估计。回忆 5.5.1 节，在时刻 t_1 和 t_2 之间，$\mathbf{x}$ 处的 MV 定义为该点从 t_1 到 t_2 的位移。我们称 t_1 时刻的帧为锚定帧，t_2 时刻的帧为目标帧①。取决于期望的应用，锚定帧在时间上可以超前或者滞后于目标帧。如图 6.4 所示，当 $t_1 < t_2$ 时，该问题称为前向运动估计；当 $t_1 > t_2$ 时，称为后向运动估计。为了标记方便，我们将使用 $\psi_1(\mathbf{x})$ 和 $\psi_2(\mathbf{x})$ 分别定义锚定帧和目标帧。通常我们可以把运动场表示为 $\mathbf{d}(\mathbf{x};\mathbf{a})$，其中，$\mathbf{a}=[a_1,a_2,\cdots,a_L]^T$ 是一个包含所有运动参数的矢量。类似地，映射函数可以定义为 $\mathbf{w}(\mathbf{x};\mathbf{a})=\mathbf{x}+\mathbf{d}(\mathbf{x};\mathbf{a})$。运动估计问题等价于估计参数矢量 $\mathbf{a}$。已经提出的方法可以归为两类：基于特征的和基于亮度的。在基于特征的方法中，首先建立两个视频帧中所选择的特征点对之间的对应关系。然后将所建立的对应关系与所选择的运动模型进行最小平方匹配，以得到运动模型参数。这种方法仅适用于参数运动模型，在确定总体运动中是相当有效的。基于亮度的方法在每个像素点应用恒定亮度假设或光流方程，要求被估计的运动尽可能地满足这个约束。这种方法比较适合于当进行的运动不能用一个简单的模型表征，并且期望一个像素或方块的运动场估计的时候。

本章只是讨论基于亮度的方法，它较广泛地应用于需要运动补偿预测和滤波的场合。一般，基于亮度的运动估计问题可以转化为一个最优化问题，这里需要回答三个关键的问题：(1) 怎样将运动场参数化？(2)用什么样的准则来估计这些参数？(3)怎样搜索这些最优参数？在这一节中，我们首先描述几种表示运动场的方法。然后介绍不同类型的估计准则。最后，我们

① 在视频编码中，当前帧 $\psi(x,y,t_1)$ 是用以前的编码帧 $\psi(x,y,t_2)$ 预测出来的，在前向运动补偿中 $t_1 > t_2$，而在后向运动补偿中 $t_1 < t_2$。在视频编码文献中 $\psi(x,y,t_2)$ 称为参考帧。

给出通常用于运动估计的搜索策略。采用不同的运动表示和估计准则的特殊的运动估计策略将在后续各节中进行介绍

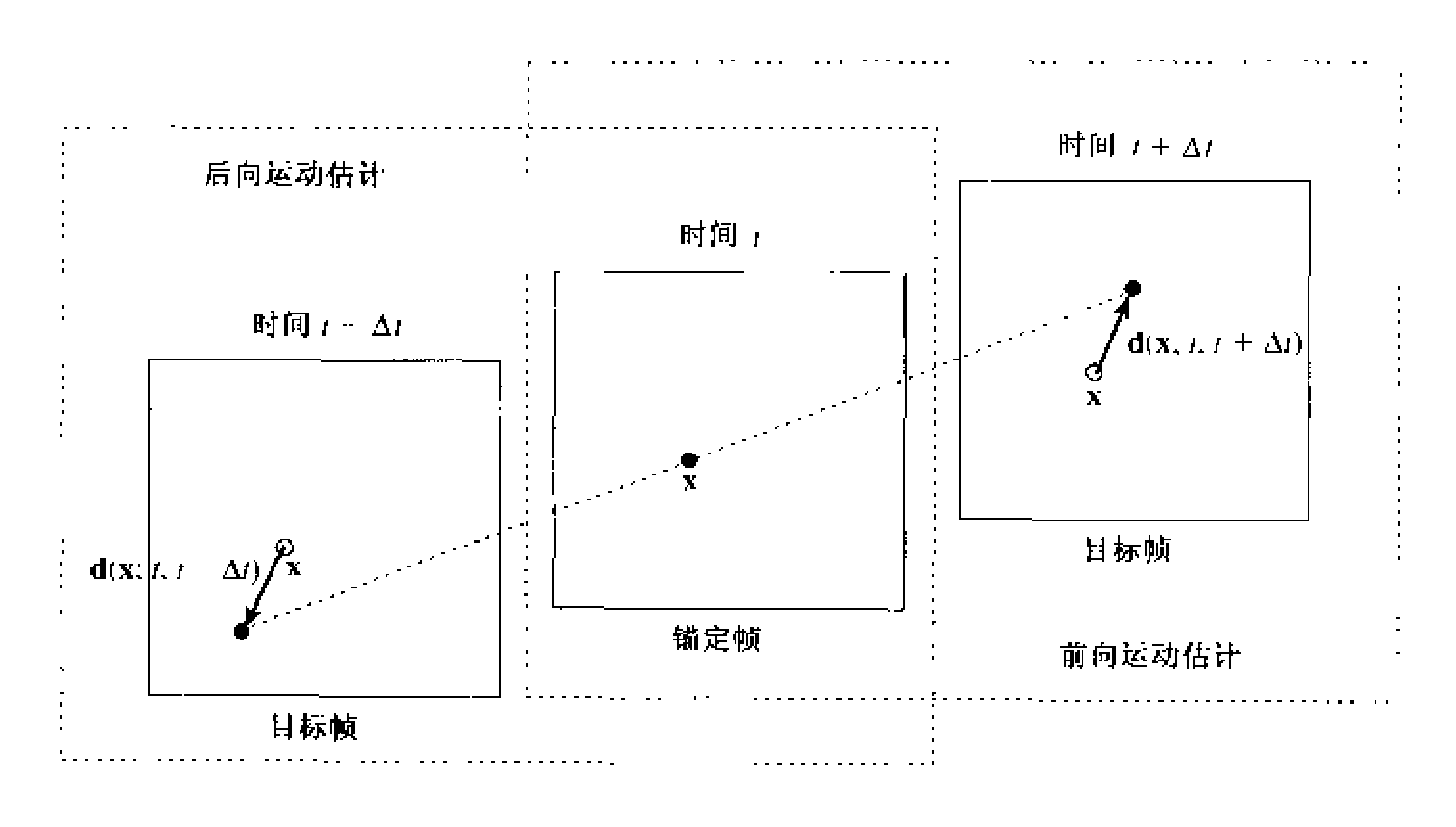


图 6.4 前向和后向运动估计

6.2.1 运动表示

运动估计的一个关键的问题是如何参数化运动场。如 5.5 节所说明的，由摄像机或物体运动产生的二维运动场通常可以用很少几个参数来描述。然而，在成像的场景中一般有多个物体做不同的运动。因此，总体参数模型通常是不适合的。最直接和不受约束的方法是在每个像素都指定运动矢量。这就是所谓基于像素表示法。这种表示法是普遍适用的，但是它需要估计大量的未知量（两倍于像素数！）并且它的解时常在物理上是不正确的，除非在估计过程中施加适当的物理约束。另一方面，若仅是摄像机在运动，或者成像景物包含单个的具有平坦表面的物体，我们可以用整体运动表示法来描述整个运动场。一般对于包含多个运动物体的景物，更适当的方法是把一个图像帧分成多个区域，使得在每个区域中的运动可以很好地用一个参数化模型表征。这被称为基于区域运动表示法①，它包括一个区域分割图和几个运动参数集，每一个集用于一个区域。这种方法的困难在于我们事先并不知道哪些像素具有相同的运动。因此，分割和估计必须用迭代完成，这需要很大的计算量，实际上可能是行不通的。

降低基于区域运动表示法的复杂性的一个方法是把图像域固定分割成许多小的方块。只要每个块足够小，则每个块内的运动变化就可以很好地用一个简单的模型表征，同时每个块的运动参数可以独立地进行估计。这就把我们带到了通常的基于块的表示法。在每一个块中模拟化运动的最简单形式是用一个常量位移，从而使估计问题变成为每一个块找到一个运动矢量(MV)。这种方法在精确度和复杂性之间提供了一个很好的折中，并且已经在实际的视频编码系统中取得了巨大的成功。基于块的方法的一个主要问题是它没有在相临块的运动过渡上

① 这有时称为基于物体运动表示法[39]。这里使用术语"基于区域"来表明这样一个事实，我们只考虑二维运动，而一个具有相关二维运动的区域不一定对应于一个物理上的物体。

施加任何约束。即使真实的运动场从一个块到另一个块是平滑变化的，也经常使运动在块的边界出现不连续现象。克服这种问题的一个方法是用基于网格的表示法，这种方法是把图像帧分割成互不重叠的多边形元素。整个帧的运动场只用节点（多边形元素的角点）的 MV 表示，而元素内部点的 MV 用节点的 MV 内插得到。这种表示法包含各处连续的运动场。它比基于块表示法更适合于一般进行连续运动的物体内部区域，但是它不能捕捉物体边界处的不连续运动。更精确的运动估计需要必要时允许不连续的自适应策略。图 6.5 给出了几种运动表示法描述一个头和肩场景的效果。在下面几节中，我们将介绍用不同运动表示法进行运动估计的方法。

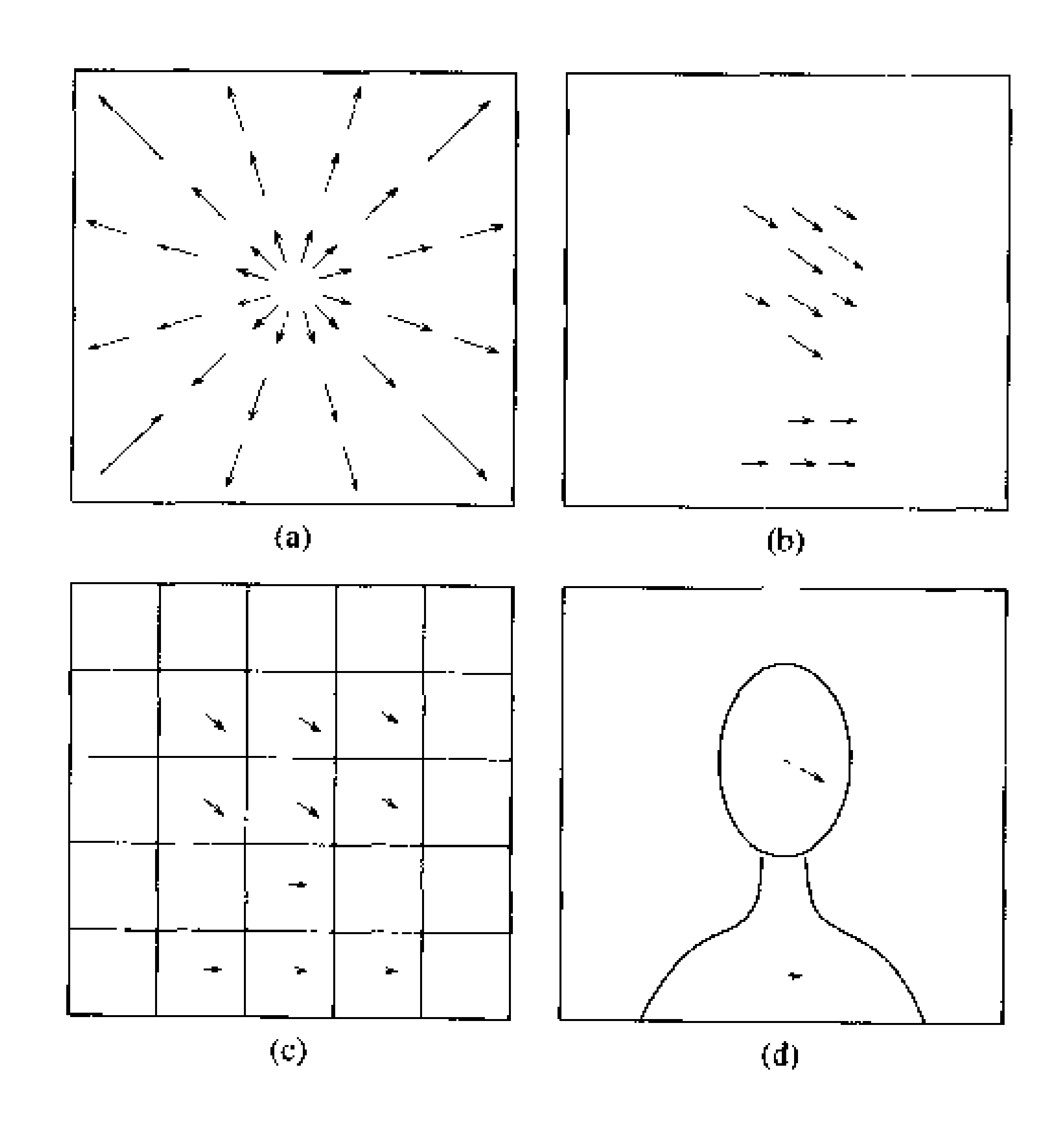


图 6.5　不同的运动表示法：(a)整体；(b)基于像素；(c)基于块；(d)基于区域

6.2.2　运动估计准则

对于一个选定的运动模型，问题是如何估计模型参数。本节将讨论几种不同的估计模型参数的准则。

基于位移帧差准则　最常用的运动估计准则是锚定帧 ψ_1 与目标帧 ψ_2 之间每个对应点对间亮度值之差的和。回忆 ψ_1 中的 $\mathbf{x}$ 移动到 ψ_2 中的 $\mathbf{w}(\mathbf{x};\mathbf{a})$。因此目标函数可以写为：

$$E_{\mathrm{DFD}}(\mathbf{a}) = \sum_{\mathbf{x}\in\Lambda} \left| \psi_2(\mathbf{w}(\mathbf{x};\mathbf{a})) - \psi_1(\mathbf{x}) \right|^p \tag{6.2.1}$$

其中 Λ 是在 ψ_1 中所有像素的集合，p 是一个正数。当 $p=1$ 时，上面的误差称为平均绝对差（MAD），当 $p=2$ 时，称为均方差（MSE）。误差图像 $e(\mathbf{x};\mathbf{a})=\psi_2(\mathbf{w}(\mathbf{x};\mathbf{a}))-\psi_1(\mathbf{x})$ 一般称为位移帧差（DFD）图像，而前面的公式用于测量 DFD 误差。

使 E_{DFD}最小的必要条件是它的梯度等于零。在 $p=2$ 的情况下,梯度是:

$$\frac{\partial E_{\mathrm{DFD}}}{\partial \mathbf{a}} = 2\sum_{\mathbf{x}\in\Lambda}(\psi_2(\mathbf{w}(\mathbf{x};\mathbf{a})) - \psi_1(\mathbf{x}))\frac{\partial \mathbf{d}(\mathbf{x})}{\partial \mathbf{a}}\nabla\psi_2(\mathbf{w}(\mathbf{x};\mathbf{a})) \tag{6.2.2}$$

其中

$$\frac{\partial \mathbf{d}}{\partial \mathbf{a}} = \begin{bmatrix} \frac{\partial d_x}{\partial a_1} & \frac{\partial d_x}{\partial a_2} & \cdots & \frac{\partial d_x}{\partial a_L} \\ \frac{\partial d_y}{\partial a_1} & \frac{\partial d_y}{\partial a_2} & \cdots & \frac{\partial d_y}{\partial a_L} \end{bmatrix}^T$$

基于光流方程准则 除了使 DFD 误差最小化以外,另外一种方法是求解基于公式(6.1.3)的光流约束建立的方程组。令$\psi_1(x,y)=\psi(x,y,t)$,$\psi_2(x,y)=\psi(x,y,t+d_t)$。若$d_t$ 很小,我们可以假设$(\partial\psi/\partial t)/dt=\psi_2(\mathbf{x})-\psi_1(\mathbf{x})$。那么公式(6.1.3)可以写成:

$$\frac{\partial\psi_1}{\partial x}d_x + \frac{\partial\psi_1}{\partial y}d_y + (\psi_2-\psi_1)=0 \text{ 或 } \nabla\psi_1^T\mathbf{d}+(\psi_2-\psi_1)=0 \tag{6.2.3}$$

这个光流方程的离散形式经常用在数字视频的运动估计中。对于所有的 $\mathbf{x}$ 解上面的方程可以转换为一个具有如下目标函数的最小化的问题:

$$E_{\mathrm{OF}}(\mathbf{a}) = \sum_{\mathbf{x}\in\Lambda}|(\nabla\psi_1(\mathbf{x}))^T\mathbf{d}(\mathbf{x};\mathbf{a})+\psi_2(\mathbf{x})-\psi_1(\mathbf{x})|^p \tag{6.2.4}$$

当 $p=2$ 时,E_{OF}的梯度为:

$$\frac{\partial E_{\mathrm{OF}}}{\partial \mathbf{a}} = 2\sum_{\mathbf{x}\in\Lambda}\Big((\nabla\psi_1(\mathbf{x}))^T\mathbf{d}(\mathbf{x};\mathbf{a})+\psi_2(\mathbf{x})-\psi_1(\mathbf{x})\Big)\frac{\partial \mathbf{d}(\mathbf{x})}{\partial \mathbf{a}}\nabla\psi_1(\mathbf{x}) \tag{6.2.5}$$

如果在一个小的区域 Λ'内运动场是一个常量,即 $\mathbf{d}(\mathbf{x};\mathbf{a})=\mathbf{d}_0,\mathbf{x}\in\Lambda'$,那么公式(6.2.5)就变成:

$$\frac{\partial E_{\mathrm{OF}}}{\partial \mathbf{d}_0} = \sum_{\mathbf{x}\in\Lambda'}\Big((\nabla\psi_1(\mathbf{x}))^T\mathbf{d}_0+\psi_2(\mathbf{x})-\psi_1(\mathbf{x})\Big)\nabla\psi_1(\mathbf{x}) \tag{6.2.6}$$

将上述梯度置为零,就得到 $\mathbf{d}_0$ 的最小平方解:

$$\mathbf{d}_0 = \Big(\sum_{\mathbf{x}\in\Lambda'}(\nabla\psi_1(\mathbf{x})(\nabla\psi_1(\mathbf{x}))^T\Big)^{-1}\Big(\sum_{\mathbf{x}\in\Lambda'}(\psi_1(\mathbf{x})-\psi_2(\mathbf{x}))\nabla\psi_1(\mathbf{x})\Big) \tag{6.2.7}$$

当运动不是常量,但与模型参数有线性关系时,我们仍可以得到类似的最小平方解(见习题6.6)。

上述方法的一个优点是,当 $p=2$ 时最小化函数是 MV 的二次函数。如果运动参数关于 MV 是线性的,那么函数具有惟一的最小值,并且容易求解。对于公式(6.2.1)给出的 DFD 误差来说并不是这样。然而,仅当运动很小或者当初始运动估计 $\bar{\mathbf{d}}(\mathbf{x})$接近能够求出的真实的运动时,光流方程是成立的,并且我们可以把$\psi_2(\boldsymbol{x})$预更新到$\psi_2(\mathbf{x}+\bar{\mathbf{d}}(\mathbf{x}))$。当不是这种情况的时候,最好应用 DFD 误差准则,用梯度下降或穷尽搜索法寻找最小解。

正则化 最小化DFD误差或解光流方程并不总是给出具有物理意义的运动估计。部分原因是恒定亮度假设不总是正确的。因为各种反射和阴影效应，同一个物体点的成像亮度在物体运动后可能发生变化。第二个原因是，在平面纹理的区域，许多不同的运动估计可以满足恒定亮度假设或光流方程。最后，如果运动参数是每一个像素的MV，光流方程就不能完全约束运动矢量。这些因素可以使得运动估计是一个病态问题。

为了得到一个有物理意义的解，我们必须施加额外的约束来正则化这个问题。常用的正则化方法是把一个代价项增加到公式(6.2.1)或(6.2.4)的误差函数中，它将强制产生的运动估计支持普通运动场的特性。一般运动场的一个众所周知的性质是，除了在物体边界处，它通常是从一个像素到一个像素平滑变化的。为了强制平滑性，可以使用一个代价项来测量邻近像素MV间的差，即：

$$E_s(\mathbf{a}) = \sum_{\mathbf{x} \in \Lambda} \sum_{\mathbf{y} \in \mathcal{N}_{\mathbf{x}}} \| d(x;a) - d(y;a) \|^2 \tag{6.2.8}$$

其中 $\|\cdot\|$ 代表二阶范数，$\mathcal{N}_{\mathbf{x}}$ 代表与 $\mathbf{x}$ 相邻的像素集合。可以使用四邻域或者八邻域[①]。

总的最小化准则可以写成：

$$E = E_{\mathrm{DFD}} + w_s E_s \tag{6.2.9}$$

权系数 w_s 应该基于与预测误差有关的运动平滑的重要性来选择。为了避免过模糊现象，应该减小物体边界的权系数。然而，这需要精确检测物体的边界，这不是一项简单的任务。

贝叶斯准则 贝叶斯准则是基于运动估计问题的概率公式，它是由Konrad和Dubois首先提出来的[22,38]。在这个公式下，给定一个锚定帧 ψ_1，在目标帧 ψ_2 的图像函数被认为是自由场 Ψ 的一个实现，而运动场 $\mathbf{d}$ 是另一个自由场 $\mathcal{D}$ 的一个实现。给定 Ψ 的一个实现和 ψ_1，应用贝叶斯准则，运动场 $\mathcal{D}$ 的后验概率分布可以写成：

$$P(\mathcal{D} = \mathbf{d} | \Psi = \psi_2; \psi_1) = \frac{P(\Psi = \psi_2 | \mathcal{D} = \mathbf{d}; \psi_1) P(\mathcal{D} = \mathbf{d}; \psi_1)}{P(\Psi = \psi_2; \psi_1)} \tag{6.2.10}$$

（在这个表示中，分号表示它后面的变量是决定性的参数。）基于贝叶斯准则的估计是要使后验概率达到最大。但是对于给定的 ψ_1 和 ψ_2，使概率最大等价于仅使分子达到最大。因此 $\mathbf{d}$ 的最大后验(MAP)估计是：

$$\mathbf{d}_{\mathrm{MAP}} = \mathrm{argmax}_{\mathbf{d}} \{ P(\Psi = \psi_2 | \mathcal{D} = \mathbf{d}; \psi_1) P(\mathcal{D} = \mathbf{d}; \psi_1) \} \tag{6.2.11}$$

第一个概率表示给出运动场的图像帧与锚定帧的似然性。令 $\mathcal{E}$ 表示对于给定的 $\mathbf{d}$ 和 ψ_1，相应于DFD图像 $e(\mathbf{x}) = \psi_2(\mathbf{x}+\mathbf{d}) - \psi_1(\mathbf{x})$ 的随机场，则：

$$P(\Psi = \psi_2 | \mathcal{D} = \mathbf{d}; \psi_1) = P(\mathcal{E} = e)$$

公式(6.2.11)变成：

$$\mathbf{d}_{\mathrm{MAP}} = \mathrm{argmax}_{\mathbf{d}} \{ P(\mathcal{E} = e) P(\mathcal{D} = \mathbf{d}; \psi_1) \}$$

① $\mathbf{x}$ 的四邻域包括 $\mathbf{x}$ 的东、南、西、北四个相邻的像素。八邻域又增加了东北、东西、东南和西南方向上的四个像素。

$$= \mathrm{argmin}_{\mathbf{d}}\{-\log P(\mathcal{E}=e)-\log P(\mathcal{D}=\mathbf{d};\psi_1)\} \tag{6.2.12}$$

第8章将讨论的信源编码理论告诉我们，信源$\mathcal{X}$的最小编码长度等于它的熵$-\log P(\mathcal{X}=x)$。我们可以看到，MAP估计准则等价于使DFD图像e与运动场$\mathbf{d}$的编码长度的和为最小。将在第9章中说明，这正是使用运动补偿预测的视频编码器必须编码的长度。因此，对于$\mathbf{d}$的MAP估计等价于最小描述长度(MDL)估计[34]。因为在视频编码中运动估计的目的是使码率达到最小，所以MAP准则与最小预测误差准则相比是一个较好的选择。

DFD图像最常用的模型是零均值独立恒等分布的(i. i. d.)高斯场，它的分布是：

$$P(\mathcal{E}=e) = (2\pi\sigma^2)^{-|\Lambda|/2}\exp\frac{-\sum_{\mathbf{x}\in\Lambda}e^2(\mathbf{x})}{2\sigma^2} \tag{6.2.13}$$

其中$|\Lambda|$代表Λ的大小(即Λ中的像素数)。使用这个模型，使公式(6.2.12)中的第一项为最小等价于使前面定义的DFD误差($p=2$时)为最小。

对于运动场$\mathcal{D}$，一个常用的模型是吉布斯/马尔可夫自由场[11]。这个模型是用称为团(clique)的邻域结构定义的。令$\mathcal{C}$代表团的集合，该模型假设：

$$P(\mathcal{D}=\mathbf{d}) = \frac{1}{Z}\exp\Big(-\sum_{c\in\mathcal{C}}V_c(\mathbf{d})\Big) \tag{6.2.14}$$

其中Z是一个归一化因子。函数$V_c(\mathbf{d})$称为潜在函数。它一般用来测量同一个团内像素之间的差：

$$V_c(\mathbf{d}) = \sum_{(\mathbf{x},\mathbf{y})\in c}|\mathbf{d}(\mathbf{x})-\mathbf{d}(\mathbf{y})|^2 \tag{6.2.15}$$

在这个模型下，最小化公式(6.2.12)中的第二项等价于最小化公式(6.2.8)中的平滑函数。因此，MAP估计等价于具有适当平滑约束的基于DFD的估计器。

6.2.3 优化方法

6.2.2节中给出的误差函数，可以用各种优化方法进行最小化。这里我们只考虑穷尽搜索和基于梯度搜索方法。一般对于穷尽搜索，MAD因为计算简单而被采用，而对于基于梯度搜索方法，采用MSE是由于数学上易于处理。

显然，穷尽搜索法的优点是它可保证达到全局最小。然而，只当未知参数的数目很小，并且每个参数只取一个有限离散值集合时，这种搜索才是可行的。为了减少搜索时间，可以开发达到次最佳解的各种快速算法。

最常用的梯度下降法包括最陡梯度下降和牛顿-拉夫森方法(这种方法一个简单的回顾在附录B给出)。基于梯度的方法能够处理高维连续空间的未知参数。然而，它只保证收敛到局部最小。前一节介绍的误差函数一般不是凸的，从而可以具有许多远离全局最小的局部最小。因此，重要的是通过使用先验知识获得一个好的初始解，或加入一个代价项使误差函数变成凸的。

对于基于梯度的方法，必须计算信号的时空梯度。附录A回顾了由数字采样图像计算一阶和二阶梯度的方法。注意，如Barron，Fleet和Beauchemin所指出的那样[4]，用于计算梯度函数的方法对于有关的运动估计方法的精确性和健壮性有着深刻的影响。使用高斯预滤波器后

接一个中心差分,一般会产生比简单的两点差分近似好得多的结果。

一种重要的搜索策略是使用运动场的多分辨率表示,并且以分层的方式进行搜索。基本思想是首先用粗精度搜索运动参数,把这个解波及到一个较细的分辨率,然后以这个较细的分辨率优化这个解。它可以克服穷尽搜索法的缓慢性和基于梯度法的非最优化。我们将在6.9节中详细地介绍多分辨率方法。

6.3 基于像素的运动估计

在基于像素的运动估计中,必须估计每一个像素的MV。显然,这个问题是难于处理的。如果使用恒定亮度假设,对于锚定帧的每一个像素,在目标帧里将会有许多具有完全相同亮度的像素。如果使用光流方程,这个问题仍然是不确定的,因为两个未知数只有一个方程。为了防止这个问题发生,一般有四种方法:第一,可以使用正则化技术在运动场上施加平滑约束,使得新像素的MV受周围像素的已找到的那些MV的约束。第二,可以假定每一个像素周围邻域中的MV是相同的,并且把恒定亮度假设或光流方程应用到整个邻域。第三,可以利用另外一些不变量约束;除导致光流方程的亮度不变量之外,我们可以假设运动中的亮度梯度是不变的,如在参考文献[29,26,15]中提出的。最后,可以利用运动前后帧的相位函数之间的关系[9]。在参考文献[4]中,Barron,Fleet和Beauchemin评价了各种光流计算方法,在合成图像和真实世界的图像上试验了这些算法。这一节只描述前两种方法。我们也介绍为视频压缩应用而开发的像素递归型算法。

6.3.1 用运动平滑约束正则化

Horn和Schunck[16]提出了通过最小化如下的目标函数来估计运动矢量,这个目标函数是基于流的准则与运动平滑性准则的联合:

$$E(\mathbf{v}(\mathbf{x})) = \sum_{\mathbf{x}\in\Lambda}\left(\frac{\partial\psi}{\partial x}v_x + \frac{\partial\psi}{\partial y}v_y + \frac{\partial\psi}{\partial t}\right)^2 + w_s(\|\nabla v_x\|^2 + \|\nabla v_y\|^2) \quad (6.3.1)$$

在他们最初的算法中,空间梯度 v_x 和 v_y 是由$\nabla v_x = [v_x(x,y)-v_x(x-1,y), v_x(x,y)-v_x(x,y-1)]^T$,$\nabla v_y = [v_y(x,y)-v_y(x-1,y), v_y(x,y)-v_y(x,y-1)]^T$ 近似的。这个误差函数的最小化是通过称为高斯-斯德尔法的基于梯度的方法实现的。

Nagle和Enkelmann对运动估计平滑约束的效果做出了一个全面的评价[30]。为了避免运动场的过平滑,Nagel建议了一个定向平滑性约束,在这种方法中,平滑性是沿着物体边界而不是穿越边界施加的[29]。这使运动估计的精度有显著的提高[4]。

6.3.2 使用多点邻域

在这种方法中,当估计像素 $\mathbf{x}_n$ 的MV时,我们假定 $\mathbf{x}_n$ 周围的一个邻域$\mathcal{B}(\mathbf{x}_n)$内的所有像素的MV都是相同的,用 $\mathbf{d}_n$ 表示。为了确定 $\mathbf{d}_n$,我们既可以最小化$\mathcal{B}(\mathbf{x}_n)$上的预测误差,又可以用最小平方法解光流方程。这里我们介绍第一种方法。为了估计 $\mathbf{d}_n$,我们最小化$\mathcal{B}(\mathbf{x}_n)$上的DFD误差:

$$E(\mathbf{d}_n) = \frac{1}{2}\sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x})(\psi_2(\mathbf{x}+\mathbf{d}_n) - \psi_1(\mathbf{x}))^2 \tag{6.3.2}$$

其中 $w(\mathbf{x})$是分配给像素 $\mathbf{x}$ 的权值(一般权值随 $\mathbf{x}$ 到 $\mathbf{x}_n$ 距离的增加而减小)。

关于 $\mathbf{d}_n$ 的梯度是:

$$\mathbf{g}(d_n) = \frac{\partial E}{\partial \mathbf{d}_n} = \sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x})e(\mathbf{x},\mathbf{d}_n)\frac{\partial \psi_2}{\partial \mathbf{x}}\bigg|_{\mathbf{x}+\mathbf{d}_n} \tag{6.3.3}$$

其中 $e(\mathbf{x},\mathbf{d}_n)=\psi_2(\mathbf{x}+\mathbf{d}_n)-\psi_1(\mathbf{x})$是在 $\mathbf{x}$ 处具有估计 $\mathbf{d}_n$ 的 DFD。令 $\mathbf{d}_n^{(l)}$代表第 l 次迭代的估计;那么一阶梯度下降法将产生如下更新算法:

$$\mathbf{d}_n^{(l+1)} = \mathbf{d}_n^{(l)} - \alpha\mathbf{g}(\mathbf{d}_n^{(l)}) \tag{6.3.4}$$

由公式(6.3.3),每次迭代的更新都依赖于在各个像素处图像梯度的和,这些梯度以在这些像素处的加权的 DFD 值为比例因子进行缩放。

我们也可以用牛顿-拉夫森方法导出一个迭代算法。由公式(6.3.3),黑塞矩阵是:

$$[\mathbf{H}(\mathbf{d}_n)] = \frac{\partial^2 E}{\partial \mathbf{d}_n^2} = \sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x})\frac{\partial \psi_2}{\partial \mathbf{x}}\left(\frac{\partial \psi_2}{\partial \mathbf{x}}^T\right)\bigg|_{\mathbf{x}+\mathbf{d}_n} + w(\mathbf{x})e(\mathbf{x},\mathbf{d}_n)\frac{\partial^2 \psi_2}{\partial \mathbf{x}^2}\bigg|_{\mathbf{x}+\mathbf{d}_n}$$

$$\approx \sum_{\mathbf{x}\in B(\mathbf{x}_n)} w(\mathbf{x})\frac{\partial \psi_2}{\partial \mathbf{x}}\left(\frac{\partial \psi_2}{\partial \mathbf{x}}\right)^T\bigg|_{\mathbf{x}+\mathbf{d}_n} \tag{6.3.5}$$

于是牛顿-拉夫森更新算法(见附录 B)为:

$$\mathbf{d}_n^{(l+1)} = \mathbf{d}_n^{(l)} - \alpha[\mathbf{H}(\mathbf{d}_n^{(l)})]^{-1}\mathbf{g}(\mathbf{d}_n^{(l)}) \tag{6.3.5}$$

这种算法比一阶梯度下降法收敛得快,但是它每一次迭代需要更多的计算。

除了使用基于梯度的更新算法以外,也可以使用穷尽搜索法寻找 $\mathbf{d}_n$,得到在一个给定搜索区域内的最小值。这将导致在 6.4.1 节将要介绍的穷尽块匹配算法。不同的是这里所使用的邻域是一个滑动窗口,每个像素的 MV 由最小化其邻域内的误差决定;一般来说邻域不一定是一个矩形块。

6.3.3 像素递归方法

在使用运动补偿预测的视频编码中,我们必须指定 MV 和 DFD 图像。对于基于像素运动表示,需要为每一个像素指定一个 MV,这是很昂贵的。在像素递归运动估计算法中,MV 是递归得出的。特别地,当前像素的 MV 是由在此之前已经编码的邻近像素的 MV 更新的。根据同样的更新规则,解码器也可以导出同样的 MV,从而 MV 不必编码。已经开发出各种这样的算法,它们的更新规则都按照统一类型的梯度下降法[31]。

尽管像素递归方法非常简单,但它们的运动估计精度相当低。结果是,预测的误差仍很大,而且需要相当多的编码比特。因为这种方法的简单性,所以它们用在较早几代的视频编解码器中。现今的编解码器使用更加复杂的运动估计算法,它们能在用于指定 MV 和 DFD 图像的比特之间提供较好的平衡;最常用的是下一节介绍的块匹配算法。

6.4 块匹配算法

正如我们所看到的,基于像素运动估计所存在的问题是,我们必须施加平滑约束以使问题正则化。在已估计的运动场上施加平滑约束的一种方法是把图像域分割成互相不重叠的称为块的小区域,并且假定每一个块内的运动都可以用一个简单的参数模型(例如恒定、仿射和双线性模型)特征化。如果块足够小,那么这种模型是相当精确的。这一节中我们介绍用这个基于块的运动表示开发出的运动估计算法。令$\mathcal{B}_m$代表第m个图像块,M代表块的数目,并且$\mathcal{M}=\{1,2,\cdots,M\}$;块的分割应该满足:

$$\bigcup_{m\in\mathcal{M}}\mathcal{B}_m=\Lambda,\qquad \mathcal{B}_m\bigcap\mathcal{B}_n=\emptyset,m\neq n$$

理论上,一个块可以具有任意的多边形形状。然而实际上几乎都用正方形(有时也使用三角形的形状,它比较适用于当每个块的运动是用仿射模型描述的时候)。

在最简单的情况下,每一个块中的运动被假定是恒定的,即整个块进行平移。这称为块平移模型。在这一节中,我们只考虑这种简单的情况,此时运动估计问题是为每一个块寻找一个MV。这种类型的算法被共同地称为块匹配算法(BMA)。在下一节中,我们将考虑较一般的情况,此时每一个块中的运动用一个较复杂的模型来表征。

6.4.1 穷尽块匹配算法

在锚定帧$\mathcal{B}_m$中给定一个图像块,此时的运动估计问题是在目标帧中确定一个匹配块$\mathcal{B}'_m$,使得这两个块之间的误差为最小。这两个块的空间位置(用中心或一个选定的角表示)之间的位移矢量$\mathbf{d}_m$是这个块的MV。在块平移模型下,$\mathbf{w}(\mathbf{x};\mathbf{a})=\mathbf{x}+\mathbf{d}_m,\mathbf{x}\in\mathcal{B}_m$,在公式(6.2.1)中的误差可以写成:

$$E(\mathbf{d}_m,\forall m\in\mathcal{M})=\sum_{m\in\mathcal{M}}\sum_{\mathbf{x}\in\mathcal{B}_m}|\psi_2(\mathbf{x}+\mathbf{d}_m)-\psi_1(\mathbf{x})|^p \tag{6.4.1}$$

因为对于一个块所估计的MV只影响这个块的预测误差,所以我们可以通过使每个块的累积预测误差最小化来个别地估计每个块的MV,即:

$$E_m(\mathbf{d}_m)=\sum_{\mathbf{x}\in\mathcal{B}_m}|\psi_2(\mathbf{x}+\mathbf{d}_m)-\psi_1(\mathbf{x})|^p \tag{6.4.2}$$

确定使这个误差为最小的$\mathbf{d}_m$的一种方法是使用穷尽搜索,这种方法被称为穷尽块匹配算法(EBMA)。如图6.6所示,EBMA为锚定帧的一个给定块$\mathcal{B}_m$确定最优$\mathbf{d}_m$的方法是,在一个预先定义的搜索区域内,把它与目标帧中所有的候选块$\mathcal{B}'_m$进行比较,并且寻找具有最小误差的一个。这两个块之间的位移就是所估计的MV。

为了减小计算量,经常使用MAD误差($p=1$)。搜索的区域一般是关于当前块对称的,左边和右边各有R_x个像素,上边和下边各有R_y个像素,如图所示。如果已知在水平和垂直方向运动的动态范围是相同的,那么$R_x=R_y=R$。估计的精度是由搜索的步长决定的,步长是相临两个候选块在水平或者垂直方向上的距离。通常,沿着两个方向使用相同的步长。在最

简单的情况下，步长是一个像素，称为整数像素精度搜索。

令块的大小是 $N \times N$ 像素，搜索范围在水平和垂直方向上都是 $\pm R$ 像素(见图 6.6)。对于一个像素的步长，锚定帧的每一个块的候选匹配块的总数是$(2R+1)^2$。令一次运算定义为包括一次减法、一次绝对值运算和一次加法。计算每一个候选估计的 MAD 的运算数是 N^2。这样，估计每一个块的 MV 的运算数是$(2R+1)^2N^2$。一个大小为 $M \times M$ 的图像有$(M/N)^2$ 个块(假定 M 是N 的倍数)。那么全部一帧的总运算数是 $M^2(2R+1)^2$。有趣的是要注意总的计算量不依赖于块的大小 N。

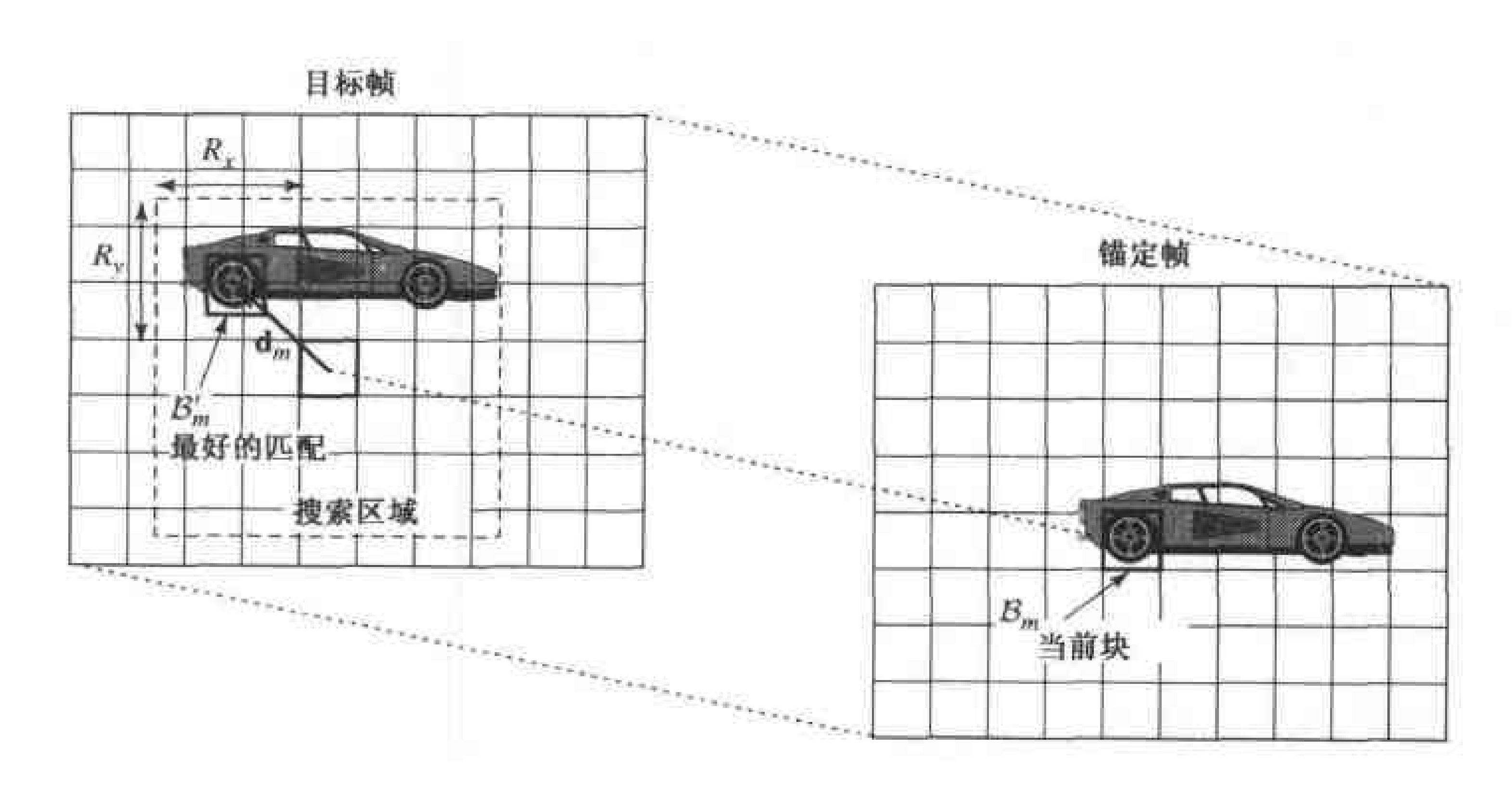


图 6.6　穷尽块匹配算法的搜索过程

举一个例子，考虑 $M=512, N=16, R=16$ 的情况：每一帧的总运算数是 2.85×10^8。对于一个帧率为 30 fps 的视频序列来说，每秒需要的运算数就是 8.55×10^9，一个惊人的数字！这个例子表明 EBMA 需要庞大的计算，这就造成了对于需要纯软件应用场合的挑战。因为这个问题，开发出了各种快速算法，用牺牲估计精度来降低计算量。在 6.4.3 节中给出了一些快速算法。EBMA 的一个优点是它可以用简单的和模块化设计的硬件实现，而且用并行的多个模块可以提高速度。许多研究致力于用 VLSI 芯片高效地实现 EBMA，有时涉及到对算法进行稍微的修改，牺牲一些精度以降低计算量、存储器空间和存储器访问时间。关于实现 EBMA 和其他快速块匹配算法的 VLSI 结构的一个综述，见参考文献[21,32,14]。

6.4.2　分数精度搜索

正如已经提示的，在 BMA 中搜索相应块的步长不一定是一个整数。为了更精确的运动表示，需要分数像素精度。使用分数步长的一个问题是对于锚定帧里给定的一个采样点，在目标帧里可能没有相应的采样点——这些样点必须由可利用的样点内插得到。通常用双线性内插达到这个目的。一般，为了实现 $1/K$ 像素步长，目标帧必须先进行 K 倍内插。图 6.7 给出了 $K=2$ 的例子，它称为半像素精度搜索。已经证明，与整数像素精度搜索相比，半像素精度搜索在估计精度上有很大的提高，特别是对于低清晰度视频。

这里自然产生的一个问题是对于运动估计合适的搜索步长是多少。显然，它依赖于所估计的运动矢量的应用场合。对于视频编码，估计的运动用于由以前已编码的帧(目标帧)预测

当前帧(锚定帧),它是应该最小化的预测误差图像(即 DFD 图像)。预测误差和搜索精度间的关系的统计分析已经由 Girod[12]考虑了,并将在第 9 章中给出。

显然,应用分数像素步长,EBMA 的复杂性进一步增加了。例如,使用半像素搜索,搜索点的总数四倍于整数像素精度搜索。考虑到内插目标帧所需的额外计算量,总的复杂性要大于四倍。

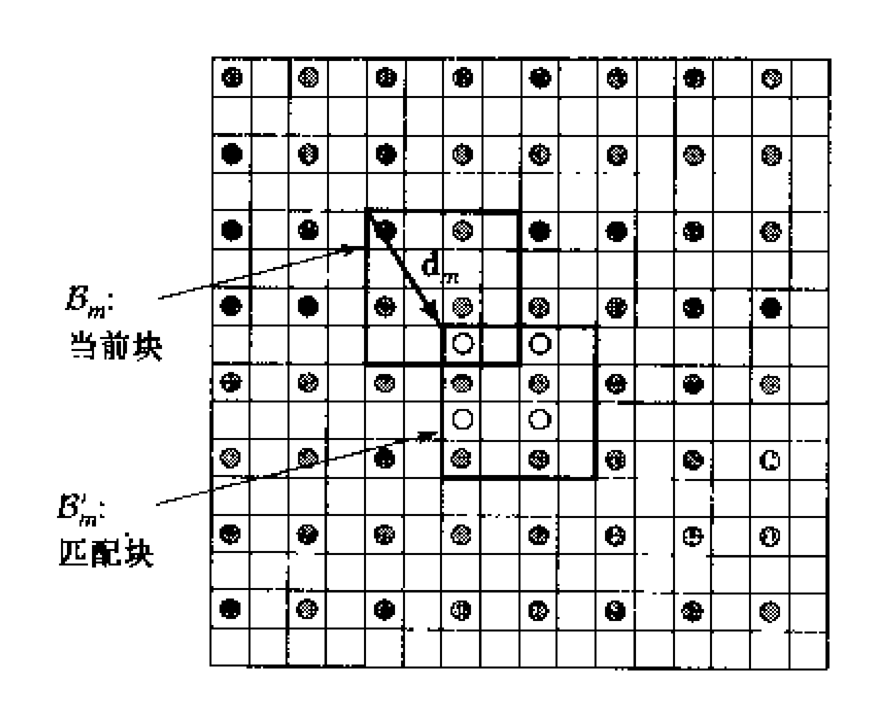


图 6.7　半像素精度块匹配。实心圆是目标帧存在的采样点,空心圆是为计算匹配误差对于候选 MV $\mathbf{d}_m = (-1, -1.5)$所要内插的样点。替换要求为每一个候选 MV 计算这些样点,一个更好的办法是对整个目标帧进行预内插

例 6.1　图 6.8(c)示出了对于图 6.8(a~b)给出的两帧,用半像素 EBMA 算法估计的运动场。图 6.8(d)示出了基于估计的运动而预测的锚定帧,这是通过用目标帧中的最佳匹配块替换掉锚定帧中的每一块得到的。图像的大小是 352×288,块的大小是 16×16。我们可以看到大量的块被精确预测;然而,也有不能很好预测的块。一些这样的块中存在非平移的运动,如那些覆盖眼睛和嘴的块。其他的块既包括前景物体又包括背景,而只有前景物体在运动。也有一些块,其图像亮度的变化是由于头部转动时,反射图形的变化引起的。这些块的运动变化不能用恒定的 MV 很好地近似,并且 EBMA 算法简单地识别那些在目标帧中与锚定帧中给定的块具有最小绝对误差的块。而且,预测的图像在某些块的边界处是不连续的,这就是通常 EBMA 算法所具有的块效应。这种效应是由块平移模型的固有限制,以及一个块的 MV 独立于它的邻块的 MV 所造成的。

预测图像和原始图像之间的精度一般是用 PSNR 测量的,如前面公式(1.5.6)所定义的。由半像素 EBMA 预测的图像的 PSNR 为 29.86 dB。对于整数像素 EBMA,得到的预测图像视觉上很相似,尽管 PSNR 稍微低一些。

6.4.3　快速算法

正如上面已经说明的,EBMA 需要很大的计算量。为了加速搜索,已经开发出各种块匹配快速算法。减少计算的关键是降低搜索候选块的数量。如前所述,对于 $\pm R$ 的搜索区域和一个像素的步长,EBMA 候选块的总数是$(2R+1)^2$。各种快速算法的不同在于它们跳过那些不

太可能有小误差的候选块。

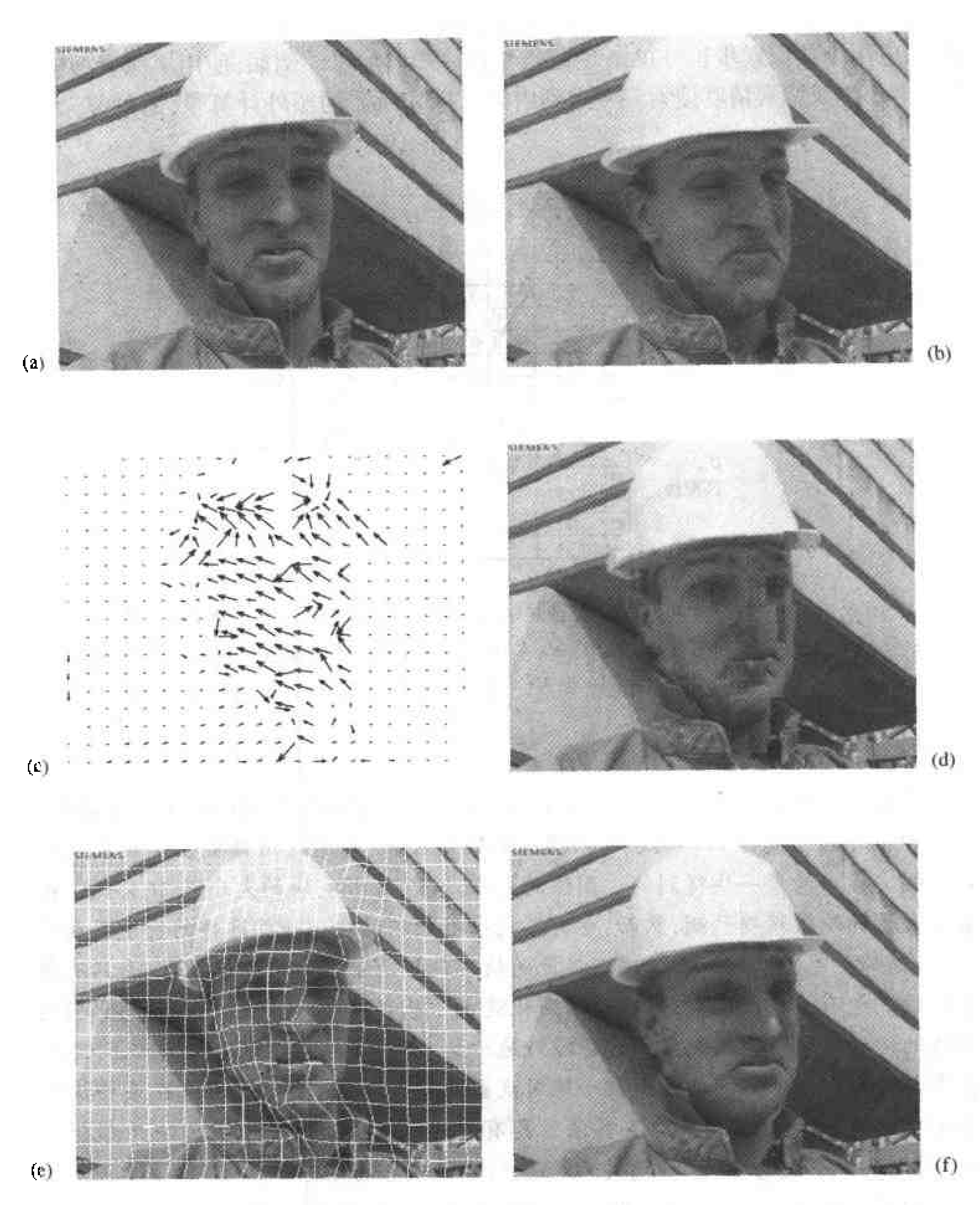


图 6.8 运动估计结果的例子:(a)目标帧;(b)锚定帧;(c~d)运动场和用半像素精度的 EBMA 得到的锚定帧的预测图像(PSNR = 29.86 dB);(e~f)运动场(用重叠在目标帧上的畸形网格表示)和用基于网格运动估计方法[43]得到的预测图像(PSNR = 29.72dB)

二维对数搜索法 一个常用的快速算法是二维对数搜索[19],如图 6.9 所示。它是从相应于零位移的位置开始搜索的。每一步试验菱形排列的五个搜索点。下一步,把中心移到前一步找到的最佳匹配点并重复菱形搜索。当最佳匹配点是中心点或是在最大搜索区域的边界上时,就减小搜索步长(菱形的半径)。否则步长保持不变。当步长减小到一个像素时,就到达了最后

一步,并且在这最后一步检验九个搜索点。初始搜索步长一般设定为最大搜索区域的一半。用这种方法,我们不能预先确定搜索步数和搜索点的总数,因为这些依赖于实际的 MV。但是最好情况(需要最少数目的搜索点)和最坏情况(需要最大数目的搜索点)是可以分析的。

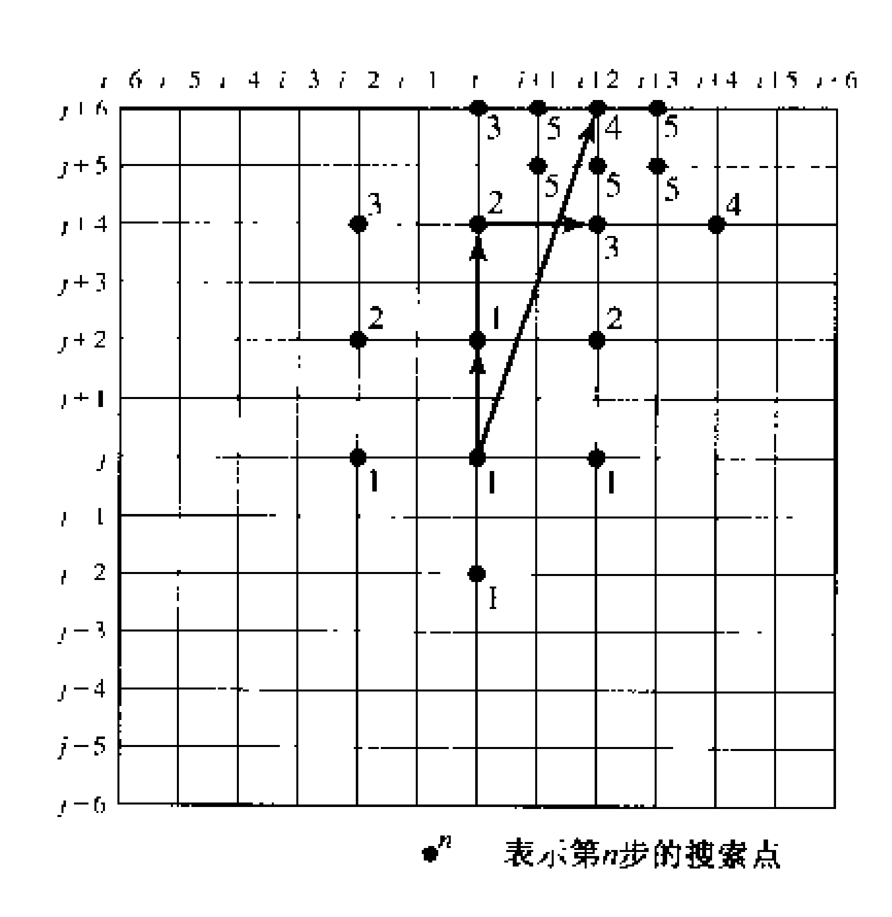


图 6.9　二维对数搜索法。目标帧的搜索点是相对于锚定帧的一个块中心(i,j)示出的。在这个例子中,1~5步中的最佳匹配MV是(0,2),(0,4),(2,4),(2,6)和(2,6)。最终的MV是(2,6)

三步搜索法　另一种常用的快速算法是三步搜索法[20]。如图 6.10 所示,这种搜索的步长从等于或者略大于最大搜索范围的一半开始。在每一步中,比较九个搜索点。它们包括搜索正方形的中心点和八个位于搜索区边界上的搜索点。每一步以后搜索步长减小一半,至搜索步长为一个像素时搜索结束。在每一个新的搜索步中,搜索中心点移到由前一步得到的最佳匹配点。令 R_0 代表初始搜索步长,那么最多有 $L=\lfloor \log_2 R_0+1 \rfloor$ 个搜索步,这里 $\lfloor x \rfloor$ 代表 x 的向下取整。如果 $R_0=R/2$,那么 $L=\lfloor \log_2 R \rfloor$。除了在一开始需要检验九个点以外,在每一个搜索步中,都搜索八个点。因此,总的搜索点数为 $8L+1$。例如,对于搜索范围 $R=32$,EBMA 的总搜索点数是 4 225,而对于三步法,点数降到 41,节约倍数超过 100。与二维对数搜索法不同,三步搜索法具有一个固定的可以预知的搜索步数和搜索点数。此外,它具有较正规的结构。这些特点使三步搜索法比二维对数法以及某些其他快速算法更适于 VLSI 的实现。

快速算法的比较　表 6.1 比较了三种不同搜索算法所需要的最小和最大搜索点数以及搜索步数。可以看出,一些算法具有较正规的结构,从而具有固定的计算数,而另一些算法具有很不相同的最好情况和最坏情况的计算数。对于 VLSI 实现,结构的正规化是很重要的,而对于软件实现,平均情况的复杂性(一般接近于最好的情况)是更重要的。关于这些和其他一些快速算法用 VLSI 电路实现的复杂性和成本的分析见参考文献[14]。

前面的讨论假定搜索精度是整数像素的。为了达到半像素精度,我们可以在任何快速算法中加入最后一步,在整数像素搜索找到的最佳匹配点的 ±1 个像素的邻域内以半像素步长进行搜索。

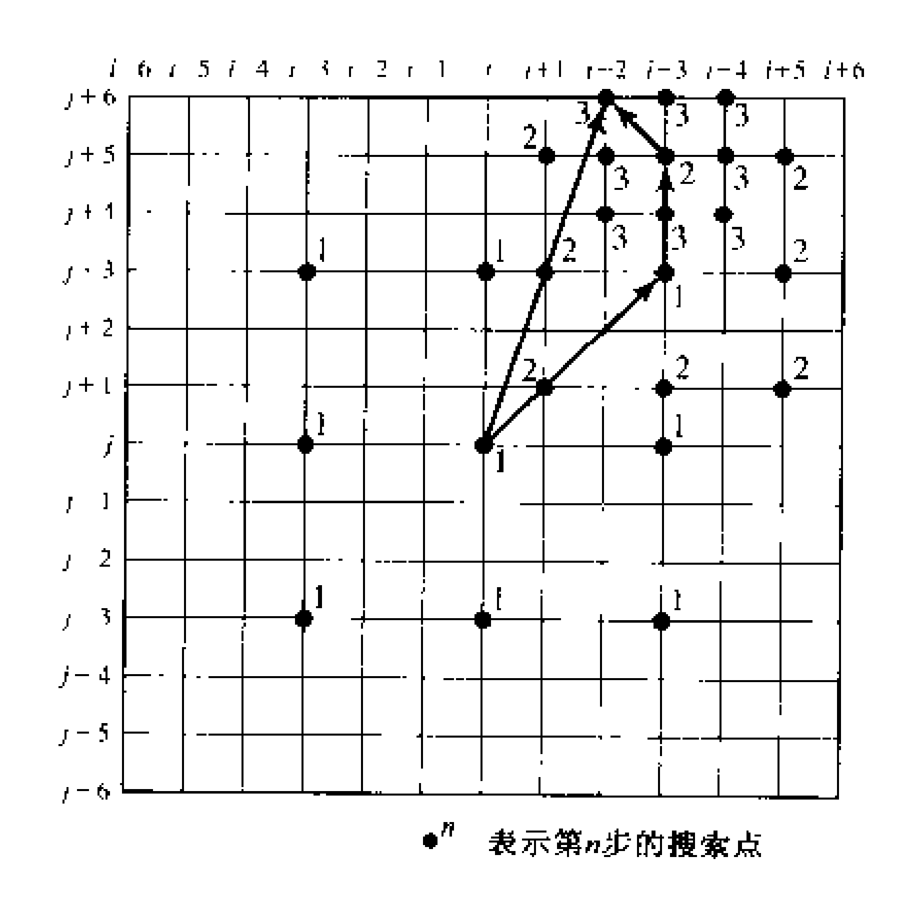


图6.10 三步搜索法。在这个例子中,1~3步的最佳匹配MV是(3,3),(3,5),(2,6),最终的MV是(2,6)

表6.1 快速搜索算法的比较($R=7$)

搜索算法	搜索点数		搜索步数	
	最 小	最 大	最 小	最 大
EBMA	225	225	1	1
2-D-log[19]	13	26	2	8
Three-step[20]	25	25	3	3

6.4.4 施加运动平滑性约束

由图6.8(c)可以看到,用EBMA获得的运动场是相当混乱的。这是因为没有对块MV的空间变化施加任何约束。已经提出了几个平滑估计的运动场的方法,使其比较接近物理的运动场。一种有效的途径是使用多分辨率方法,它首先用粗空间分辨率估计MV,然后不断地以相继较精细的分辨率优化MV。MV由较粗分辨率向较细分辨率的过渡是由空间内插完成的,它使产生的运动场具有一定程度的空间连续性(这个技术将在6.9节中进行更加详细的讨论)。另一种方法是明确地施加一个平滑性约束,把一个平滑项加到测量相邻块的MV变化的公式(6.4.2)的误差准则中。除了运动矢量是定义在块上和预测误差需要在块上求和以外,产生的总的误差函数将类似于公式(6.2.9)。难题是确定预测误差项与平滑项之间适当的权重,使得到的运动场不至于过平滑。理想情况下,加权应该是自适应的:在接近物体边界时不应使用。一个更困难的任务是确定物体的边界,在那里应允许运动的不连续。

6.4.5 相位相关法*

除了最小化DFD以外,另外一种运动估计的方法是识别相位相关函数中的峰值。假定两

* 标有星号的节可以跳过或者留待进一步研究。

个图像帧是纯平移的关系，那么：

$$\psi_1(\mathbf{x}) = \psi_2(\mathbf{x}+\mathbf{d}) \tag{6.4.3}$$

两边取傅里叶变换并利用傅里叶位移定理，我们得到：

$$\Psi_1(\mathbf{f}) = \Psi_2(\mathbf{f}) \cdot e^{j2\pi\mathbf{d}^T\mathbf{f}} \tag{6.4.4}$$

$\psi_1(\mathbf{x})$和$\psi_2(\mathbf{x})$之间的归一化互功率谱是：

$$\tilde{\Psi}(\mathbf{f}) = \frac{\Psi_1(\mathbf{f}) \cdot \Psi_2^*(\mathbf{f})}{|\Psi_1(\mathbf{f}) \cdot \Psi_2^*(\mathbf{f})|} = e^{j2\pi\mathbf{d}^T\mathbf{f}} \tag{6.4.5}$$

其中上标*表示复共轭。取逆傅里叶变换得到相位相关函数(PCF)①：

$$\mathrm{PCF}(\mathbf{x}) = \mathcal{F}^{-1}\{\tilde{\Psi}(\mathbf{f})\} = \delta(\mathbf{x}+\mathbf{d}) \tag{6.4.6}$$

我们看到两个互为平移的图像的 PCF 是一个冲激函数，冲激位于正好等于两个图像间平移的位置。通过确定 PCF 的峰值位置，可以估计两个图像之间的平移。这个方法首先是由 Kuglin [23]为图像配准所使用的。

前面的推导假定图像在空间连续并且是无限大小的。然而，实际的图像信号是离散的和有限的。实际上，在可利用的图像域应用 DSFT，相当于 CSFT 在无限图像域上周期性扩展，而 CSFT 在给定的图像域外为零(见 2.1 节及 2.2 节)。为了抑制由于采样造成的混叠效应，在计算公式(6.4.5)时经常使用频率域加权函数 $W(\mathbf{f})$(在参考文献[13]中，使用 $\alpha=0.2$ 的凯塞窗作为加权函数)。为了减小边界采样效应，在计算 DSFT 之前，也可以对$\psi_1(\mathbf{x})$和$\psi_2(\mathbf{x})$应用一个空间域加权函数 $w(\mathbf{x})$。

上述相位相关法广泛应用于图像配准，这时全部图像都必须对准[33]，[10]。对于运动估计，两幅图像一般不是总体位移的关系。为了处置这种情况，相位相关函数经常应用在块级。对于大小为 $N\times N$ 且不相重叠的块的运动估计，两个帧一般都分割为 $L\times L$ 大小的不重叠的区域块。对于 $\pm R$ 的搜索区域，区域块的尺寸应该为 $L\geq N+2R$。为了确定$\psi_1(\mathbf{x})$中块的 MV，将一个尺寸为 $L\times L$ 的离散傅里叶变换(DFT)应用于这个块和在$\psi_2(\mathbf{x})$中与它相应的块。然后用相同尺寸的逆 DFT 计算 PCF，并确定峰值的位置。为了应用快速傅里叶变换(FFT)算法，L 一般选为 2 的幂。例如，若 $N=16, R=16$，则 $L=64$ 是合适的。

这种方法假定相应的两个区域块之间有一个总体平移。对于一般的视频序列这个假定不成立。当在$\psi_1(\mathbf{x})$的区域块中有几个进行不同运动的小块时，我们将在 PCF 中看到几个峰值。每个峰值对应于一个小块的运动。峰值的位置指示该块的 MV，而峰值的幅度正比于小块的大小[40]。在这个意义上，PCF 揭示了与块上 MV 的直方图类似的信息。为了估计块中起支配作用的 MV，我们首先提取 PCF 的局部极大值。然后检测相应的 MV 的 DFD。产生最小 DFD 的 MV 将被认为是该块的 MV。由于只有少量的候选 MV 被检测，与全搜索方法相比，可以显著降低计算复杂性。

这种方法可以扩展到分数像素精度的运动估计。在参考文献[13]中，通过改变候选运动矢量的长度到 ±1 个像素，整数像素候选运动矢量被增大了。在参考文献[37]和[40]中提出了其他的方法。

① 这个名称来自这样的事实：它是函数$\psi_1(\mathbf{x})$和$\psi_2(\mathbf{x})$的相位部分的互相关。

用于运动估计的相位相关法的优点是它对亮度的变化不敏感(见5.2节)。这是因为图像均值的变化,或者图像乘以一个常数,并不影响相位信息。对基于DFD的方法这就不成立了。

6.4.6 二进制特征匹配*

在这种所谓分层特征匹配运动估计策略(HFM-ME)[25]中,与用于传统块匹配方法中的像素亮度值相对应,定义了符号截断特征(STF)并应用于块匹配。应用STF定义,数据块用一个均值和一个二进制比特图形表示。块匹配运动估计被分解为均值匹配和二进制相位匹配。与EBMA相比,这个技术使计算的复杂度显著降低,因为二进制相位匹配只涉及布尔逻辑运算。使用STF也显著减少了帧缓冲器和运动估计器之间的数据传送时间。实验表明在同样的搜索区域下,HFM-ME可以达到类似于EBMA的预测精度,但它的实现可以快约64倍。当HFM-ME的搜索区域加倍时,它比EBMA的预测更加精确,同时仍具有可观的时间节省[25]。

一个大小为$2^N\times 2^N$的块的STF矢量包括两个部分。第一部分由多分辨率均值矢量组成,第二部分由符号截断二进制矢量组成。均值矢量的递归定义如下:

$$\text{Mean}^n(i,j)=\left\lfloor \frac{1}{4}\sum_{p=0}^{1}\sum_{q=0}^{1}\text{Mean}^{n+1}(2i+p,2j+q)\right\rfloor,\quad 0\leqslant i,j\leqslant 2^n-1$$

$$0\leqslant n\leqslant N-1$$

$$\text{Mean}^N(i,j)=\psi(i,j),\qquad 0\leqslant i,j\leqslant 2^N-1 \quad (6.4.7)$$

其中$\{\psi(i,j);0\leqslant i,j\leqslant 2^N-1\}$是原始块的像素亮度值。

符号截断矢量由下式得到:

$$\text{ST_pattern}^n(i,j)=\begin{cases}0 & \text{如果 } \text{Mean}^n(i,j)\geqslant \text{Mean}^{n-1}\left(\left\lfloor\frac{i}{2}\right\rfloor,\left\lfloor\frac{j}{2}\right\rfloor\right)\\ 1 & \text{其他}\end{cases} \quad (6.4.8)$$

于是,一个$2^N\times 2^N$的块分解到第n级的STF矢量可以表示成:

$$\text{STFV}_n^N=\{\text{ST_pattern}^N,\text{ST_pattern}^{N-1},\cdots,\text{ST_pattern}^{N-n+1},\text{mean}^{N-n}\} \quad (6.4.9)$$

当$n=N$时,一个块被完全分解,具有如下的STF矢量:

$$\text{STFV}_N^N=\{\text{ST_pattern}^N,\text{ST_pattern}^{N-1},\cdots,\text{ST_pattern}^1,\text{mean}^0\} \quad (6.4.10)$$

所有中间的均值矢量只用于生成ST图形(ST_pattern),并且可以舍弃。因此,最终的STF表示包括一个具有$\frac{4}{3}(4^N-1)$比特的多分辨率二进制序列以及一个字节的均值。与原来4^N字节的像素值相比,它给出了一个减小很多的数据集。同时,这个特征集允许在块匹配中运用二进制的布尔操作。

作为一个例子,让我们考虑如何对于一个4×4数据块构造两层的STF。首先,构造均值塔如下:

$$\begin{bmatrix}158 & 80 & 59 & 74\\ 80 & 69 & 59 & 74\\ 87 & 86 & 65 & 62\\ 116 & 100 & 72 & 58\end{bmatrix}\Rightarrow\begin{bmatrix}97 & 67\\ 97 & 64\end{bmatrix}\Rightarrow 81$$

---

* 标有星号的节可以跳过或者留待进一步研究。

然后得到如下的 STF 矢量：

$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}, 81$$

上述例子的 STF 矢量分解到一层的 STF 矢量为{0110 1110 1101 0001,(97,67,97,64)}。完全分解的 STF 矢量为{0110 1110 1101 0001,0101,81}。它包括一个 20 比特的二进制图形，其中包括一个 4×4 的第一层 ST 图形，一个 2×2 的第二层 ST 图形和一个均值。在实际实现中，完全分解的 STF 矢量或者混合层的 STF 矢量都可以使用。

两个 STF 矢量的比较是由两个并行的判决过程完成的：(1)计算均值之间的绝对误差，(2)确定两个二进制图形之间的汉明距离。后者可以用 XOR 布尔运算非常快地完成。因此，HFM-ME 的主要计算负荷是计算当前的和所有候选匹配块的均值塔。然而计算可以提前做，对于每一个可能的块仅做一次。关于它的计算复杂性的详细分析和使用对数搜索的快速算法见参考文献[25]。

6.5 可变形块匹配算法

在前面介绍的块匹配算法中，假定每一个块进行单纯的平动。这种模型不适用于块的旋转、变焦等。一般可以用较复杂的模型，例如仿射、双线性或投影映射描述每一个块的运动(显然，这仍将包括作为特例的平动模型)。对于这种模型，一个锚定帧中的块一般被映射到一个非正方的四边形，如图 6.11 所示。因此，我们把这种使用高阶模型的基于块的运动估计方法称为可变形块匹配算法(DBMA)[24](也称为广义块匹配算法[36])。下面，我们首先讨论怎样只用块角(称为节点)处的 MV 内插出块中任何点的 MV，然后给出一个估计节点 MV 的算法。

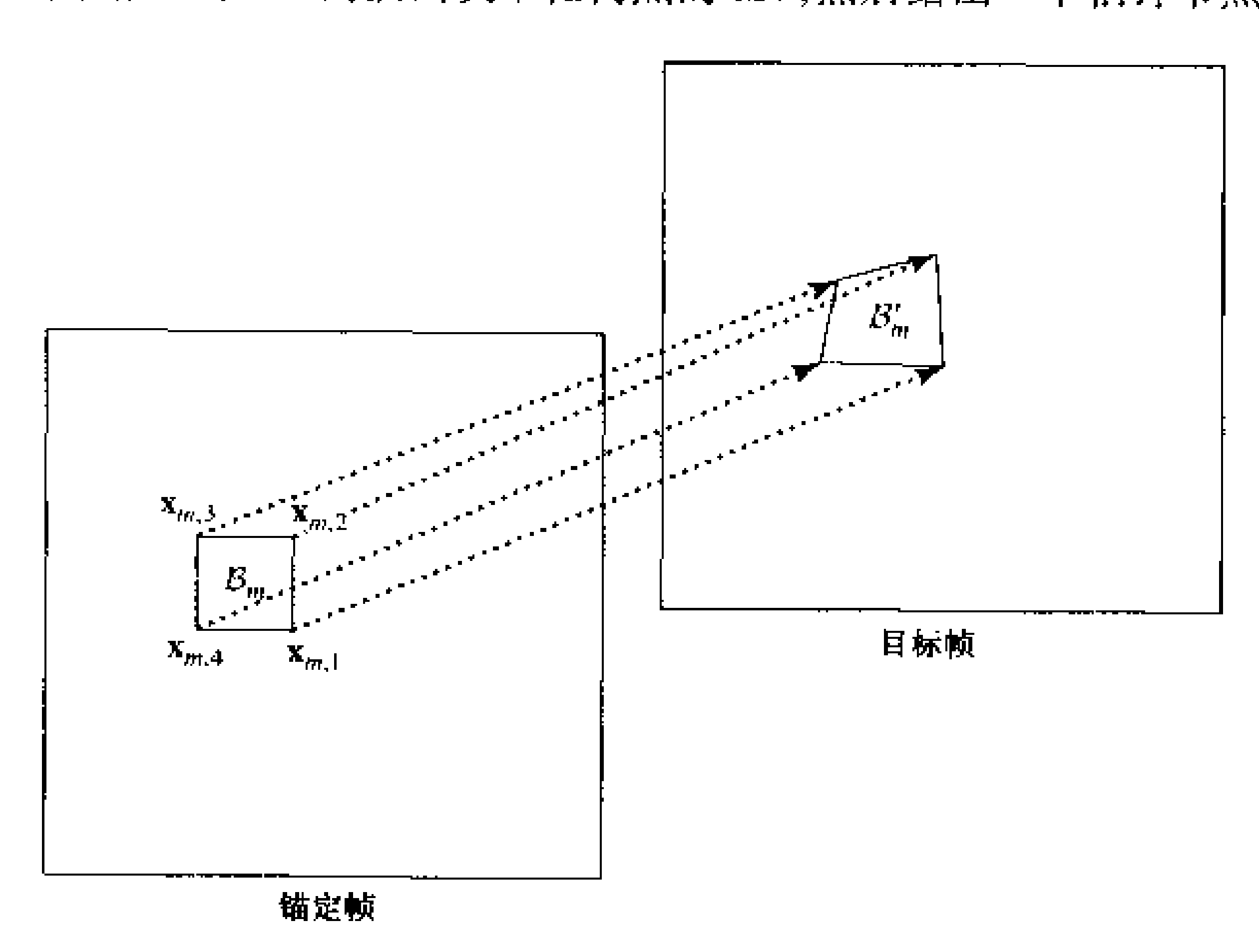


图 6.11 可变形块匹配算法为锚定帧中的每一个块寻找在目标帧中最佳匹配的四边形。允许的块变形依赖于块所使用的运动模型

6.5.1 基于节点的运动表示

在5.5节中,我们描述了几个相应于不同三维运动的二维运动模型。所有这些模型都可以用来表征块中的运动。在5.5.4节中,我们说明了怎样用不同阶的多项式映射来近似最一般的模型——投影映射。在本节中将介绍一个基于节点的块运动模型,它可以表征与多项式模型同样的运动,但易于说明和规定。

在这个模型中,我们假设在块中所选择的某个数目的控制节点能够自由运动,而且任何中间点的位移可以通过节点的位移内插。令 K 表示控制节点数,$\mathbf{d}_{m,k}$ 表示 $\mathcal{B}_m$ 中控制节点的MV,则块的运动函数可描述为:

$$\mathbf{d}_m(\mathbf{x}) = \sum_{k=1}^{K} \phi_{m,k}(\mathbf{x})\mathbf{d}_{m,k}, \quad x \in \mathcal{B}_m \tag{6.5.1}$$

公式(6.5.1)表示作为节点位移的内插块中任意点的位移,如图6.12所示。内插核 $\phi_{m,k}(\mathbf{x})$ 取决于 $\mathcal{B}_m$ 中第 k 个控制点所期望的对 $\mathbf{x}$ 的贡献。一种设计这种内插核的方法是使用与相应的节点结构有关的形状函数(在6.6.1节中我们进一步讨论形状函数的设计)。

前面介绍的平移、仿射和双线性模型分别是基于节点的模型在一、三和四个节点时的特殊情况。具有更多节点的模型可以表征更复杂的变形。在单节点情况下(在块中心或一个选择的角)的内插核是一个脉冲函数,相当于最近邻内插。在三节点(块的任意三个角)和四节点(四个角)情况下的内插函数分别是仿射函数和双线性函数。要对矩形块应用仿射模型,首先要把矩形分成两个三角形,然后每一个三角形模型化为三节点模型。

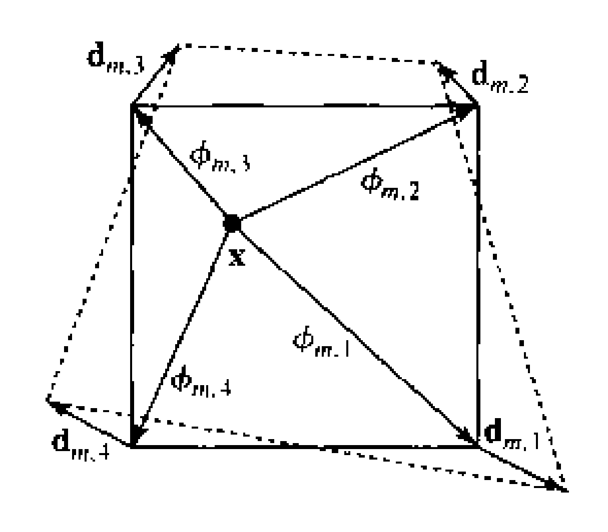


图6.12 在块中通过节点MV的运动内插

与前面介绍的基于多项式表示法相比较,基于节点的表示法易于可视化。给出双线性函数的8个系数,能想像出可变形块的模样吗? 但是如果给出块的四个角点的位置,则一定能做到! 而且,节点的MV可以较容易地估计,并且可以用比多项式系数低的精度规定它。根据关于运动的动态范围和期望估计精度的先验知识,确定节点MV合适的搜索范围和搜索步长比确定多项式系数更容易。除此以外,基于节点表示法中的所有运动参数都是同等重要的,但在多项式表示中这些参数不能同等对待(例如,高阶系数的估计要比常数项困难得多)。最后,规定多项式系数需要很高的精度:高阶系数一个小的变化可能产生很不相同的运动场。另一方面,为了规定节点的MV,整数或者半像素精度通常就够用了。这些特点对于视频编码应用是很重要的。

6.5.2 采用基于节点模型的运动估计

因为从块到块的节点运动估计是独立的,所以我们省略了用来标明哪个块正在被考虑的脚标 m。下面的推导应用于任何块$\mathcal{B}$。对于基于节点的运动模型,任何块的运动参数是节点的 MV;即 $\mathbf{a}=[\mathbf{d}_k,k\in\mathcal{K}]$,其中$\mathcal{K}=\{1,2,\cdots,K\}$。它们可以通过使这个块上的预测误差最小化来进行估计,即:

$$E(\mathbf{a})=\sum_{\mathbf{x}\in\mathcal{B}}|\psi_2(\mathbf{w}(\mathbf{x};\mathbf{a}))-\psi_1(\mathbf{x})|^p \tag{6.5.2}$$

其中

$$\mathbf{w}(\mathbf{x};\mathbf{a})=\mathbf{x}+\sum_{k\in\mathcal{K}}\phi_k(\mathbf{x})\mathbf{d}_k \tag{6.5.3}$$

与 BMA 一样,有很多方法来最小化公式(6.5.2)中的误差,包括穷尽搜索和各种基于梯度的搜索方法。然而,需要较大计算量的穷尽搜索法,在实际中可能因为高维搜索空间而是不可接受的。而基于梯度的搜索算法在这种情况下是切实可行的。下面,我们按参考文献[24]中的方法推导牛顿-拉夫森搜索算法。

定义 $\mathbf{a}=[\mathbf{a}_x^T,\mathbf{a}_y^T]^T$,其中 $\mathbf{a}_x=[d_{x,1},d_{x,2},\cdots,d_{x,K}]^T$,$\mathbf{a}_y=[d_{y,1},d_{y,2},\cdots,d_{y,K}]^T$。可以证明:

$$\frac{\partial E}{\partial\mathbf{a}}(\mathbf{a})=\begin{bmatrix}\frac{\partial E}{\partial\mathbf{a}_x}(\mathbf{a})\\ \frac{\partial E}{\partial\mathbf{a}_y}(\mathbf{a})\end{bmatrix}$$

其中

$$\frac{\partial E}{\partial\mathbf{a}_x}(\mathbf{a})=2\sum_{\mathbf{x}\in\mathcal{B}}e(\mathbf{x};\mathbf{a})\frac{\partial\psi_2(\mathbf{w}(\mathbf{x};\mathbf{a}))}{\partial x}\phi(\mathbf{x})$$
$$\frac{\partial E}{\partial\mathbf{a}_y}(\mathbf{a})=2\sum_{\mathbf{x}\in\mathcal{B}}e(\mathbf{x};\mathbf{a})\frac{\partial\psi_2(\mathbf{w}(\mathbf{x};\mathbf{a}))}{\partial y}\phi(\mathbf{x})$$

在上面的公式中,$e(\mathbf{x};\mathbf{a})=\psi_2(\mathbf{w}(\mathbf{x};\mathbf{a}))-\psi_1(\mathbf{x})$和 $\phi(\mathbf{x})=[\phi_1(\mathbf{x}),\phi_2(\mathbf{x}),\cdots,\phi_K(\mathbf{x})]^T$。

舍掉涉及二阶梯度的项,黑塞矩阵可以近似为:

$$[\mathbf{H}(\mathbf{a})]=\begin{bmatrix}[\mathbf{H}_{xx}(\mathbf{a})] & [\mathbf{H}_{xy}(\mathbf{a})]\\ [\mathbf{H}_{xy}(\mathbf{a})] & [\mathbf{H}_{yy}(\mathbf{a})]\end{bmatrix}$$

其中

$$[\mathbf{H}_{xx}(\mathbf{a})]=2\sum_{\mathbf{x}\in\mathcal{B}}\left(\frac{\partial\psi_2}{\partial x}\right)^2\Big|_{\mathbf{w}(\mathbf{x};\mathbf{a})}\phi(\mathbf{x})\phi(\mathbf{x})^T$$
$$[\mathbf{H}_{yy}(\mathbf{a})]=2\sum_{\mathbf{x}\in\mathcal{B}}\left(\frac{\partial\psi_2}{\partial y}\right)^2\Big|_{\mathbf{w}(\mathbf{x};\mathbf{a})}\phi(\mathbf{x})\phi(\mathbf{x})^T$$

$$[\mathbf{H}_{xy}(\mathbf{a})] = 2\sum_{\mathbf{x}\in\mathcal{B}}\frac{\partial\psi_2}{\partial x}\frac{\partial\psi_2}{\partial y}\Big|_{\mathbf{w}(\mathbf{x};\mathbf{a})}\phi(\mathbf{x})\phi(\mathbf{x})^T$$

牛顿-拉夫森更新算法是：

$$\mathbf{a}^{(l+1)} = \mathbf{a}^{(l)} - \alpha[\mathbf{H}(\mathbf{a}^{(l)})]^{-1}\frac{\partial E}{\partial\mathbf{a}}(\mathbf{a}^{(l)}) \quad (6.5.4)$$

每次迭代中的更新都需要 $2K\times 2K$ 对称矩阵[$\mathbf{H}$]的求逆。

为了减少数字计算，我们可以分别更新 x 和 y 方向上的位移。通过类似的推导将得到：

$$\mathbf{a}_x^{(l+1)} = \mathbf{a}_x^{(l)} - \alpha[\mathbf{H}_{xx}(\mathbf{a}^{(l)})]^{-1}\frac{\partial E}{\partial\mathbf{a}_x}(\mathbf{a}^{(l)}) \quad (6.5.5)$$

$$\mathbf{a}_y^{(l+1)} = \mathbf{a}_y^{(l)} - \alpha[\mathbf{H}_{yy}(\mathbf{a}^{(l)})]^{-1}\frac{\partial E}{\partial\mathbf{a}_y}(\mathbf{a}^{(l)}) \quad (6.5.6)$$

在这种情况下，每次更新只需要两个 $K\times K$ 矩阵的求逆。在四节点的情况下，[$\mathbf{H}$]是一个 8×8 矩阵，而[$\mathbf{H}_{xx}$]和[$\mathbf{H}_{yy}$]是 4×4 矩阵。

与所有的基于梯度的迭代过程一样，如果初始解选择得不合适，上述更新算法可能达到一个坏的远离全局最小的局部最小。EBMA 通常可以提供一个好的初始解。例如，考虑四节点模型，每一个块的每一个角都是一个节点。可以用与每个节点相连的四个块的运动矢量的平均作为这个节点 MV 的初始估计。然后这个初始估计可以用公式(6.5.4)相继地更新。

要注意，这种算法也可以用于基于多项式的运动表示。其中，$\mathbf{a}_x$ 和 $\mathbf{a}_y$ 分别代表与水平和垂直位移有关的多项式系数，而 $\phi_k(\cdot)$对应于基本的多项式基函数。然而，为 $\mathbf{a}_x$ 和 $\mathbf{a}_y$ 设定搜索区域以及检验所得到的运动场的可行性是困难的。

6.6 基于网格的运动估计*

对于 BMA 或者 DBMA 所使用的基于块的模型，各个块中的运动参数都是独立规定的。

除非邻近块的运动参数被约束得非常平滑，所估计的运动场通常是不连续，有时还是混乱的，如图 6.13(a)所示。解决这个问题的一个办法是采用基于网格的运动估计。如图 6.13(b)所示，锚定帧被一个网格所覆盖，运动估计的问题是寻找每一个节点的运动，使得锚定帧每一个元素内的图案与目标帧中相应的变形元素很好地匹配。每一个元素内的运动是由节点的 MV 内插得到的。只要目标帧的节点保持构成一个可行的网格，基于网格的运动表示就保证是连续的，从而不会有与基于块的表示相关联的块失真。基于网格表示的另外一个优点是，它能够连续地跟踪相继帧上的相同的节点集，这在需要物体跟踪的应用中是所希望的。如图 6.13(c)所示，我们可以为初始帧生成一个网格，然后再在每两个帧间估计节点的运动。每一个新的帧(锚定帧)都使用前一步产生的网格，使得相同的节点集在所有的帧得到跟踪。这对于基于块的表示是不可能的，因为它要求每一个新的锚定帧被重置为包含常规块的一个分割。

---

* 标有星号的节可以跳过或者留待进一步研究。

应该注意，基于网格表示的内在连续性不总是所希望的。能用这种表示捕捉的运动类型，可以形象化为橡胶板的变形，它是各处连续的。在真实世界的视频序列中，物体边界处的运动经常是不连续的。更精确的表示可以对不同的物体使用分离的网格。与基于块的表示一样，基于网格的表示的精度依赖于节点数。只要使用足够数量的节点，就可以重现非常复杂的运动场。为了使所需要的节点数最小化，网格应该自适应成像场景，使每个元素中的真实运动是平滑的(即可以由节点的运动精确地内插)。如果使用一个常规的、非适应的网格，那么为了精确地近似运动场就需要大量的节点。

下面，我们首先讲述怎样用基于网格的表示指定一个运动场，然后提出估计网格中的节点运动的算法。

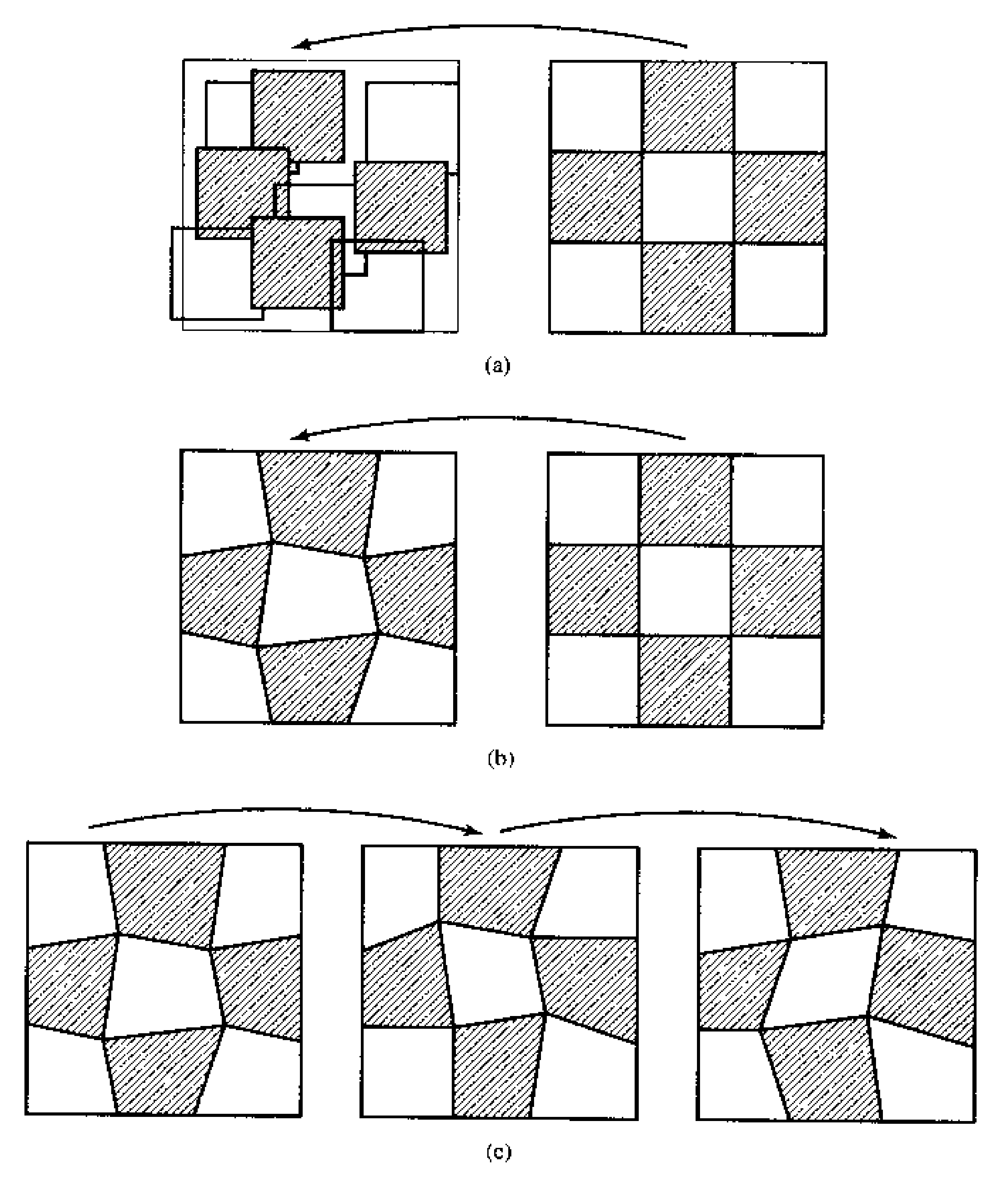


图 6.13　基于块和基于网格的运动表示法比较：(a)两帧之间基于块的运动估计，在锚定帧中的每一个块内采用平移模型；(b)两帧之间基于网格的运动估计，锚定帧采用常规网格；(c)基于网格的运动跟踪，每一个新的锚定帧采用跟踪的网格

6.6.1 基于网格的运动表示

在基于网格的运动表示中，锚定帧中的图像域被分割成互不重叠的多边形元素。每一个元素用几个节点和节点间的连线表示，如图 6.14 所示。这样的网格也称为控制网格。在基于网格的表示中，整个帧的运动场只用节点处的 MV 描述。一个元素内部点的 MV 由该元素节点的 MV 内插得到。节点的 MV 是受约束的，使目标帧中的节点保持构成可行的网格，没有卷边的元素。

令 M 和 N 分别表示元素数和节点数，K 表示定义每个元素的节点数。为了方便，我们定义下面的索引集：$\mathcal{M}=\{1,2,\cdots,M\}$，$\mathcal{N}=\{1,2,\cdots,N\}$，$\mathcal{K}=\{1,2,\cdots,K\}$。帧 t（$t=1$ 表示锚定帧；$t=2$ 表示目标帧）的第 m 个元素和第 n 个节点用 $\mathcal{B}_{t,m}$，$m\in\mathcal{M}$ 和 $\mathbf{x}_{t,n}$，$n\in\mathcal{N}$ 表示，第 n 个节点的 MV 用 $\mathbf{d}_n=\mathbf{x}_{2,n}-\mathbf{x}_{1,n}$ 表示。元素 $\mathcal{B}_{1,m}$ 中的运动场与节点 $\mathbf{d}_n$ 的关系是：

$$\mathbf{d}_m(\mathbf{x})=\sum_{k\in\mathcal{K}}\phi_{m,k}(\mathbf{x})\mathbf{d}_{n(m,k)},\quad \mathbf{x}\in\mathcal{B}_{1,m}\tag{6.6.1}$$

其中 $n(m,k)$ 指定第 m 个元素中的第 k 节点的全局索引（见图 6.14）。函数 $\phi_{m,k}(\mathbf{x})$ 是与元素 m 中的节点 k 有关的内插核。它决定了 $\mathcal{B}_{1,m}$ 中的第 k 个节点对 $\mathbf{x}$ 处的 MV 期望的贡献。这种内插机制已示于前面的图 6.12。为了保证穿越元素边界的连续性，内插核应该满足：

$$0\leqslant\phi_{m,k}(\mathbf{x})\leqslant 1,\quad \sum_k\phi_{m,k}(\mathbf{x})=1,\quad \forall\mathbf{x}\in\mathcal{B}_{1,m}$$

和

$$\phi_{m,k}(\mathbf{x}_l)=\delta_{k,l}=\begin{cases}1 & k=l\\0 & k\neq l\end{cases}$$

在有限元法（FEM）分析中，这些函数称为形状函数[45]。如果所有的元素具有同样的形状，那么所有的形状函数就是相等的，即 $\phi_{m,k}(\mathbf{x})=\phi_k(\mathbf{x})$。

标准的三角形和四边形元素示于图 6.15。标准三角形元素的形状函数是：

$$\phi_1^t(x,y)=x,\quad \phi_2^t(x,y)=y,\quad \phi_3^t(x,y)=1-x-y\tag{6.6.2}$$

标准四边形元素的形状函数是：

$$\begin{aligned}&\phi_1^q(x,y)=(1+x)(1-y)/4,\quad \phi_2^q(x,y)=(1+x)(1+y)/4\\&\phi_3^q(x,y)=(1-x)(1+y)/4,\quad \phi_4^q(x,y)=(1-x)(1-y)/4\end{aligned}\tag{6.6.3}$$

我们看到这两种情况的形状函数分别是仿射和双线性函数。关于任意三角形元素的形状函数，读者可以参阅参考文献[41]。这些函数的系数依赖于节点位置。

应该注意的是，公式（6.6.1）中每一个元素内的运动表示与公式（6.5.1）介绍的基于节点的运动表示相同，除了节点和元素是用全局索引表示以外。这是必需的，因为从一个元素到一个元素节点的 MV 不是独立的。重要的是不要把基于网格的模型与上一节介绍的基于节点的模型混淆起来。尽管几个相邻的块可能共享相同的节点，但是每个块的节点 MV 是独立确定的。回到图 6.14（b），在基于网格的模型中，节点 n 被分配一个单独的 MV，它将影响与这个节点相连的四个四边形元素中内插的运动函数。在基于节点的模型中，节点 n 可以具有四个不

同的 MV,这取决于要考虑哪个块的运动。

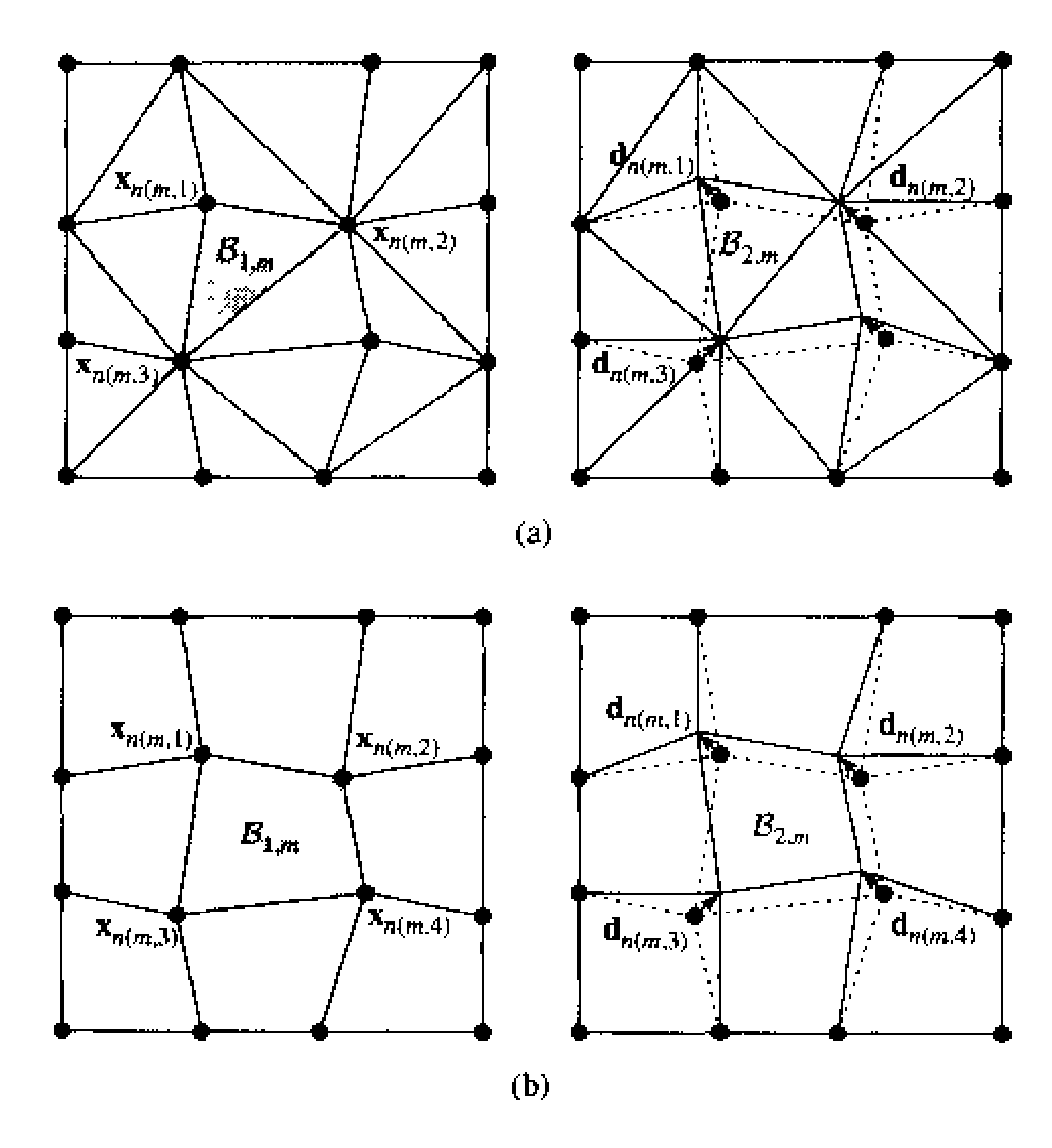


图 6.14　基于网格运动表示的图解:(a)使用三角形网格;每一个元素有三个节点相连;(b)使用四边形网格,每一个元素有四个节点相连。在这个例子中,两个网格有同样数量的节点,但三角形网格中有两倍的元素。左边的一列表示锚定帧初始的网格,右边的一列是目标帧中的变形网格

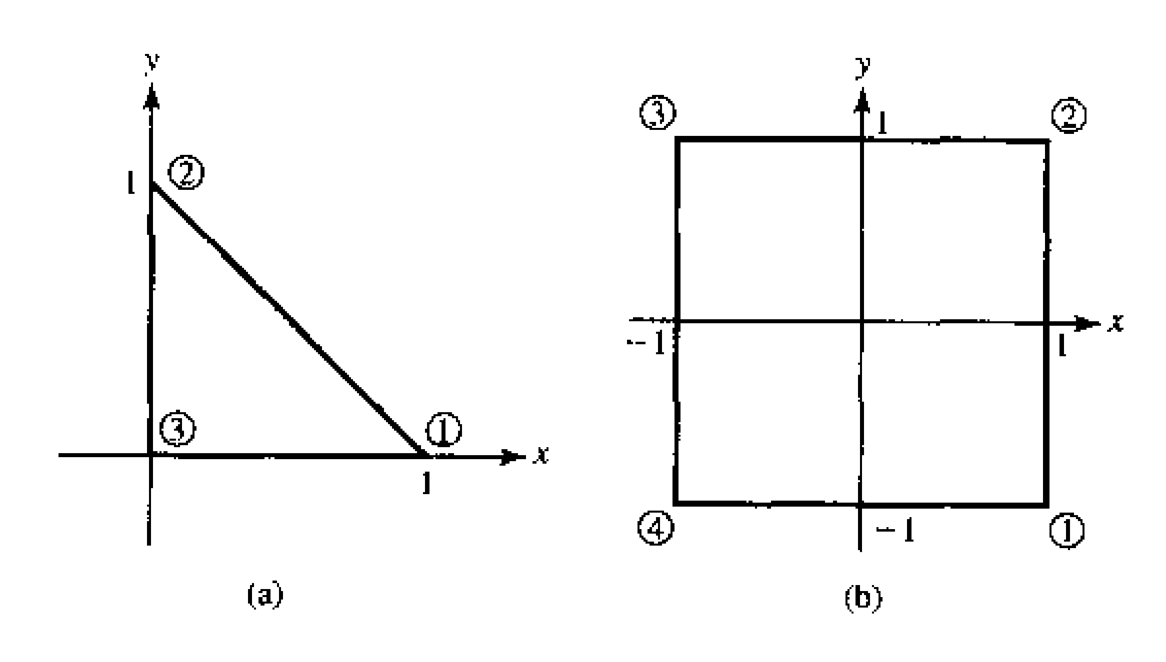


图 6.15　(a)一个标准三角形元素;(b)一个标准四边形元素(正方形)

6.6.2 采用基于网格模型的运动估计

对于基于网格的运动表示,一般要解决两类问题:(1) 在锚定帧中给定一个网格(或者等效地,节点),如何确定目标帧中的节点位置——这实质上是一个运动估计问题。(2) 在锚定帧中如何建立网格,使得网格与物体的边界一致。注意,每个元素对应于单个物体的一小块光滑表面的网格比任意配置的网格(例如,正规网格)能得到更精确的运动估计。一个物体自适应的网格也将更适于帧序列的运动跟踪。本书中我们只是考虑第一个问题,关于网格生成问题,见参考文献[42,3]。

对于由公式(6.6.1)描述的基于网格的运动表示,运动参数包括节点 MV,即 $\mathbf{a}=\{\mathbf{d}_n, n\in\mathcal{N}\}$。为了估计它们,可以再次使用误差最小化方法。在基于网格的模型下,公式(6.2.1)中的 DFD 误差变为:

$$E(\mathbf{d}_n, n\in\mathcal{N}) = \sum_{m\in\mathcal{M}}\sum_{\mathbf{x}\in\mathcal{B}_{1,m}} \left|\psi_2(\mathbf{w}_m(\mathbf{x})) - \psi_1(\mathbf{x})\right|^p \tag{6.6.4}$$

其中,按照公式(6.6.1)有:

$$\mathbf{w}_m(\mathbf{x}) = \mathbf{x} + \sum_{k\in\mathcal{K}} \phi_{m,k}(\mathbf{x})\mathbf{d}_{n(m,k)}, \quad \mathbf{x}\in\mathcal{B}_{1,m}$$

一般,因为$\mathcal{B}_{1,m}$的形状不规则,公式(6.6.4)的误差函数难于计算。为了简化计算,我们可以认为$\mathcal{B}_{t,m}, t=1,2$是由一个具有规则形状的主元素变形得到的。一般情况下不同元素的主元素可能是不同的。这里我们只考虑这种情况,就是所有的元素都具有相同的拓扑结构,这个拓扑结构可以由同一个主元素$\tilde{\mathcal{B}}$映射得到。图 6.16 显示了这种映射。

令 $\tilde{\phi}_k(\mathbf{u})$表示与$\tilde{\mathcal{B}}$中第 k 个节点有关的形状函数;那么由$\tilde{\mathcal{B}}$到$\mathcal{B}_{t,m}$的映射函数可以表示为:

$$\tilde{\mathbf{w}}_{t,m}(\mathbf{u}) = \sum_{k\in\mathcal{K}} \tilde{\phi}_k(\mathbf{u})\mathbf{x}_{t,n(m,k)}, \quad \mathbf{u}\in\tilde{\mathcal{B}}, \quad t=1,2 \tag{6.6.5}$$

相应于标准三角形和四边形元素的形状函数已在公式(6.6.2 及 6.6.3)中给出了。公式(6.6.4)中的误差可以在主元素上计算得到:

$$E(\mathbf{d}_n, n\in\mathcal{N}) = \sum_{m\in\mathcal{M}}\sum_{\mathbf{u}\in\tilde{\mathcal{B}}} \left|\tilde{e}_m(\mathbf{u})\right|^p \left|J_m(\mathbf{u})\right| \tag{6.6.6}$$

其中

$$\tilde{e}_m(\mathbf{u}) = \psi_2(\tilde{\mathbf{w}}_{2,m}(\mathbf{u})) - \psi_1(\tilde{\mathbf{w}}_{1,m}(\mathbf{u})) \tag{6.6.7}$$

表示这两个图像帧之间在那些都是由主元素(见图 6.16)中的 $\mathbf{u}$ 映射而来的点上的误差。函数 $J_m(\mathbf{u})$表示变换 $\tilde{\mathbf{w}}_{1,m}(\mathbf{u})=[\mathbf{w}_x(\mathbf{u})\ \mathbf{w}_y(\mathbf{u})]^T$ 的雅可比行列式[①]:

① 严格地讲,只当误差用$\tilde{\mathcal{B}}$上的积分定义时,雅可比行列式才是正确的。这里假设用求和来近似积分时,$\tilde{\mathcal{B}}$上的采样足够密。

$$J_m(\mathbf{u}) = \det\left(\frac{\partial \widetilde{\mathbf{w}}_{1,m}(\mathbf{u})}{\partial \mathbf{u}}\right) = \det\begin{bmatrix} \dfrac{\partial \mathbf{w}_x}{\partial \mathbf{x}} & \dfrac{\partial \mathbf{w}_y}{\partial x} \\ \dfrac{\partial \mathbf{w}_x}{\partial y} & \dfrac{\partial \mathbf{w}_y}{\partial y} \end{bmatrix} \tag{6.6.8}$$

$$= \left(\sum_{k\in K} \frac{\partial \tilde{\phi}_k}{\partial x} x_n\right)\left(\sum_{k\in K} \frac{\partial \tilde{\phi}_k}{\partial y} y_n\right) - \left(\sum_{k\in K} \frac{\partial \tilde{\phi}_k}{\partial x} y_n\right)\left(\sum_{k=K} \frac{\partial \tilde{\phi}_k}{\partial_y} x_n\right)$$

其中 $\mathbf{u} = [x\ y]^T$，$\mathbf{x}_{1,n(m,k)} = [x_n, y_n]^T$。

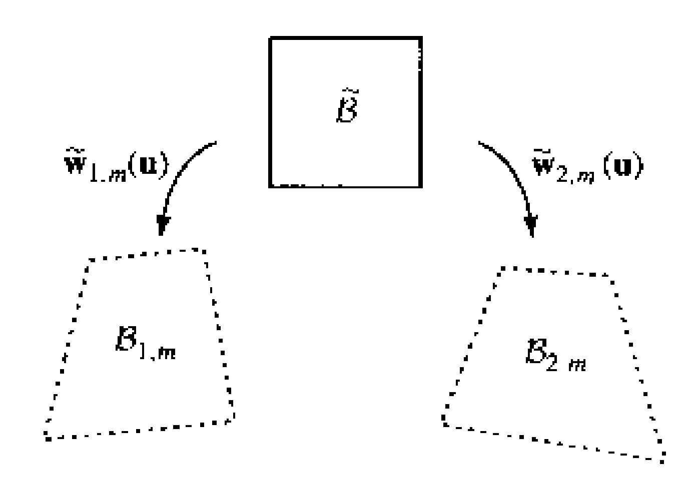


图 6.16　从主元素 $\widetilde{B}$ 到锚定帧和目标帧中的两个相应元素 $B_{1,m}$ 和 $B_{2,m}$ 的映射

对于一系列帧的运动跟踪，因为每一个新锚定帧使用的网格都是由前一步产生的跟踪网格，所以 $B_{1,m}$ 的形状一般是不规则的(见图 6.13(c))。于是，映射函数 $\widetilde{\mathbf{w}}_1(\mathbf{u})$ 和雅可比行列式 $J_m(\mathbf{u})$ 依赖于 $B_{1,m}$ 中的节点位置。另一方面，对于两帧间的运动估计，为了减小复杂性，我们可以对锚定帧使用正规网格，使每个元素本身等于主元素(见图 6.13(b))。在这种情况下，我们有 $\mathbf{w}_{1,m}(\mathbf{u}) = \mathbf{u}$ 和 $J_m(\mathbf{u}) = 1$。

当 $p = 2$ 时，公式(6.6.6)中误差函数的梯度是：

$$\frac{\partial E_p}{\partial \mathbf{d}_n} = 2\sum_{k\in K}\sum_{\mathbf{u}\in\widetilde{B}} \bar{e}_{m(n,k)}(\mathbf{u})\tilde{\phi}_k(\mathbf{u}) \left.\frac{\partial \psi_2(\mathbf{x})}{\partial \mathbf{x}}\right|_{\widetilde{\mathbf{w}}_{2,m(n,k)(\mathbf{u})}} \left|J_{m(n,k)}(\mathbf{u})\right| \tag{6.6.9}$$

其中 $m(n,k)$ 指定与第 n 个节点相连的第 k 个元素的全局索引。图 6.17 示出了在四边形网格情况下连接到节点 n 的相邻元素和形状函数。

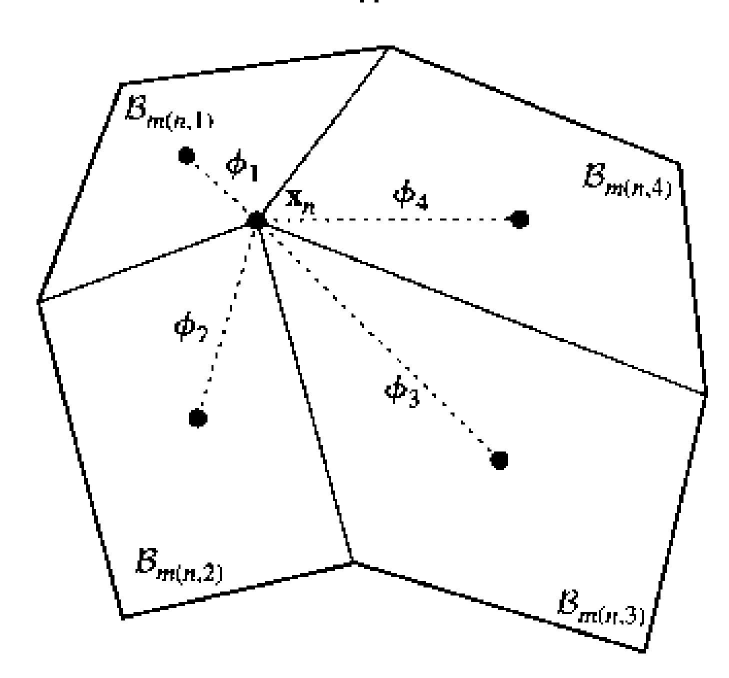


图 6.17　四边形网格的邻域结构：对于给定的节点 n，有四个与它相连的元素，每一个元素具有一个与这个节点相连的形状函数

可以看出，关于一个节点的梯度仅依赖于与它相连的几个元素中的误差。理想情况下，在基于梯度的搜索算法的每一次迭代中，为了计算任意节点的梯度函数，应假定其他的节点是固定在前一次迭代得到的位置上。一旦迭代结束，在进行下一次迭代前这些节点都应被更新。但实际上，为了加速过程，我们可以一次更新一个节点，同时固定它周围的节点。当然，这种次优化的方法会导致发散或者收敛到一个局部最小，它比由同时更新所有节点所得到的值要差。除了用通常的光栅顺序更新节点外，为了提高精度和收敛速度，我们可以对节点排序，使运动矢量能够被较精确估计的节点先得到更新。因为平滑区域运动估计的不确定性，所以最好先更新具有大的边缘幅度和小的运动补偿误差的节点。这称为最高置信度优先[7]，这种方法取自参考文献[2]。另外一种可能是把所有的节点分成几组，使同一组中的节点不共有同一个元素，因此它们对误差函数的影响是独立的。顺序地更新同一组中的节点就等价于同时更新这些节点(这是节选自参考文献[42]的方法)。可以使用一阶梯度下降法或二阶牛顿-拉夫森型的更新算法。二阶算法收敛要快得多，但它更易于收敛到坏的局部最小。

新更新的基于梯度函数的节点位置可能导致过度变形的元素(包括卷边的和钝角的元素)。为了避免发生这种情况，我们应该限制更新的节点位置可以落入的搜索范围。如果更新的位置超出了这个范围，那么就应该把它们投影到定义的搜索区域中最近的点。图 6.18 示出了四边形网格情况下的合理的搜索区域。

前面的讨论不仅应用于基于梯度的更新算法，而且应用于穷尽搜索算法。在这种情况下，我们可以一次更新一个节点，搜索的节点位置将使图 6.18 所示的搜索区域内与它相连的元素中的预测误差最小化。对每一个候选位置，用公式(6.6.6)计算误差，但是只在与该节点相连的元素上进行累积。最优位置是具有最小误差的位置。同样，在这里搜索顺序是很重要的。

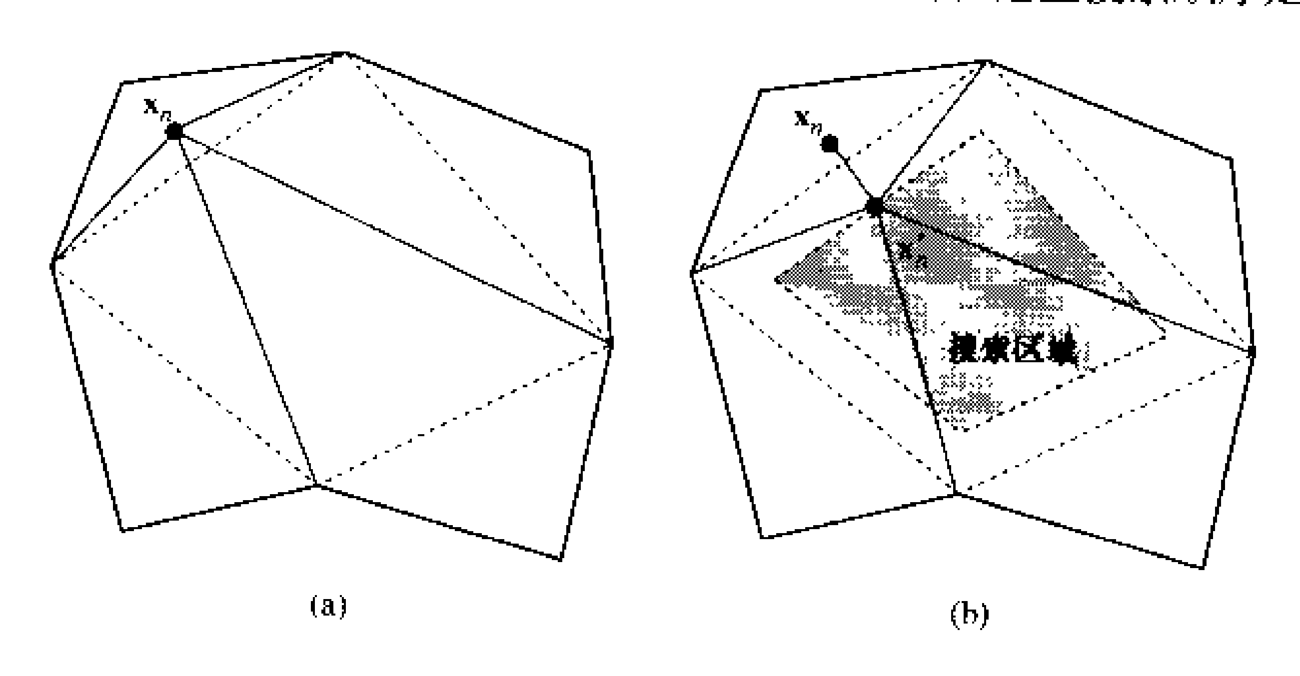


图 6.18 在给定其他节点的位置时，节点 n 的搜索区域：菱形区域(虚线)是理论极限，内部菱形区域(阴影)是实际使用的。当 $\mathbf{x}_n$ 落在如图(a)所示的菱形区域的外面的时候，至少有一个与它相连的元素是钝角的。通过把 $\mathbf{x}_n$ 投影到如图(b)中所示的内部菱形，四个元素都不会过度变形

例 6.2 图 6.8(e)和(f)示出了在每一个新帧使用矩形网格用穷尽搜索法进行后向运动估计获得的结果[43]。图 6.8(e)是覆盖在目标帧上部的变形网格，图 6.8(f)是锚定帧的预测图像。注意，图 6.8(e)中每一个变形的四边形相应于锚定帧中的一个正方形块。在脸的

右边的一个窄四边形说明它在锚定帧中被扩展了。可以看到，相应于平滑的运动场，网格也是平滑变形的。预测图像并不会出现与 EBMA 有关的块失真(图 6.8(d)和(f))，表现出的是原始图像的一个较成功的预测。然而在预测图像(图 6.8(f))与原始图像(图 6.8(b))之间仔细地比较可以看出，闭眼睛和嘴运动没有被精确地预测，同时在下巴和脖子附近有人为的扭曲失真。实际上，预测图像的 PSNR 低于由 EBMA 获得的 PSNR。

至此，我们假定为整个当前帧产生(或者在前向跟踪情况下，由前一帧推演)单一的网格，这个网格中的每一个节点被目标帧中的一个并且只有一个节点所跟踪，因此目标帧的节点仍形成一个覆盖整个帧的网格。为了处理场景中新出现和消失的物体，应该允许删除对应于消失物体的节点，以及建立新出现物体的新节点。关于这个问题的解释见参考文献[3]。

6.7 全局运动估计

在 5.5 节中我们指出，根据摄像机和物体运动以及物体表面的几何形状，相同成像物体的两个图像间的运动场可以用平移、几何变换、仿射映射和投影映射来描述。如果整个运动场是由摄像机运动引起的，或者如果成像场景包含一个进行三维刚性运动的物体，那么这样的模型就可以应用在整个帧中[①]。

实际上，我们很难寻找一个包含单一物体的视频序列。通常最少有两个物体：一个静止的背景和一个运动的前景。更经常的情况是有一个以上的前景物体。幸好，当前景物体的运动与摄像机的运动相比很小，并且摄像机不是沿 Z 方向运动时，运动场可以用一个全局模型更好地近似。例如，当摄像机以一个相对快的速度在场景中摇摄，或变焦到一个特殊的对象时就是这种情况。这种摄像机的运动在体育运动视频和电影中是相当常见的。即使当真实的二维运动场不能用一个单一的整体运动表示时，只要摄像机的运动效果与其他运动(个别小物体的运动)相比占主导地位，确定这个主导的全局运动就仍是非常有用的。本节讨论全局运动估计。

一般有两种估计全局运动的方法。一种是在一个给定运动参数集下，通过最小化预测误差来直接估计全局运动参数。另外一种方法是首先用前面描述的技术确定像素或块的运动矢量，然后用一种回归方法寻找与估计的运动场最匹配的全局运动模型。后一种方法也可以应用于所选定的特征点(例如具有很强的边缘的点)的运动矢量。

6.7.1 健壮估计器

估计全局运动的一个困难是一个像素可能不只是经历全局运动。通常，任何像素的运动可以分解为全局运动(由摄像机运动引起)和由运动物体引起的局部运动。因此，即使利用正确的全局运动参数，单独使用全局运动模型得到的估计误差可能也不是很小的。另一种情况是，不是同一帧的所有像素都经历全局运动，理想情况下我们不应该对整帧应用相同的运动模型。当全局运动与其他局部运动相比占主导地位时，这些问题可以用健壮估计法[15]予以克服。占主导地位的意思是，经历相同的全局运动，或者经历这个全局运动的像素与那些不经历这个全局运动的像素相比占有图像域中相当大的一部分。

① 回忆当摄像机或物体在 Z 方向运动的情况下，只有物体表面是平坦的，运动场才能够用投影映射表示(见 5.5.3 节)。当物体表面是空间变化时，任何点的映射函数也依赖于该点的表面深度，并且不能用全局模型表示。

健壮估计的基本思想是认为被全局运动主宰的像素是内围层,剩下的像素是外露层。起初,我们假定像素都进行相同的全局运动,并且通过最小化所有像素的预测或匹配误差来估计运动参数。这将产生一个初始的运动参数集。然后用这个初始解,我们可以计算每个像素的预测或匹配误差。误差超过某一阈值的像素被归为外露层,并在下一次迭代时去掉。然后对剩下的内围层像素重复这个过程,直到没有外露层像素存在为止。这种方法称为硬阈值健壮估计器。

若不用在每次迭代的末尾简单地把一个像素分类为内围层或是外露层的方法,我们也可以为每个像素设计一个不同的权重,对小的误差加大的权值,否则反之。在下一个最小化或匹配迭代中,使用一个加权的误差测量,从而在前一次迭代中有较大误差的像素将比那些有较小误差的像素具有较少的影响。这个方法称为软阈值健壮估计器。

6.7.2 直接估计

无论在硬阈值或者软阈值健壮估计器中,每一次迭代都涉及误差函数的最小化。这里我们导出当模型参数是直接由最小化预测误差得到时该函数的形式。由于硬阈值的情况可以认为是权值为 0 或 1 时的特例,我们只考虑软阈值的情况。令从锚定帧到目标帧的运动场用 $\mathbf{d}(\mathbf{x};\mathbf{a})$表示,其中 $\mathbf{a}$ 是包含所有全局运动参数的矢量,预测误差可以根据公式(6.2.1)写为:

$$E_{\mathrm{DFD}} = \sum_{n\in\mathcal{N}} w_n \mid \psi_2(\mathbf{x}_n + \mathbf{d}(\mathbf{x}_n;\mathbf{a})) - \psi_1(\mathbf{x}_n) \mid^p \tag{6.7.1}$$

其中,$\mathbf{x}_n, n\in\mathcal{N} = \{1,2,\cdots,N\}$ 表示估计全局运动所用的所有像素,w_n 是像素 $\mathbf{x}_n$ 的加权系数。在健壮估计过程的每一次迭代中,参数矢量 $\mathbf{a}$ 是通过最小化这个误差,用基于梯度或穷尽搜索法估计出来的。加权因子 w_n 在一次新的迭代中,将会根据 $\mathbf{x}_n$ 的 DFD 进行调整,DFD 是根据前一次迭代中估计的运动参数计算的。

6.7.3 间接估计

对于间接估计,我们假定运动矢量 $\mathbf{d}_n$ 已经在一组足够密集的点 $\mathbf{x}_n, n\in\mathcal{N}$ 估计出来了。例如,这可以使用前面所述的基于块或基于网格的方法实现。我们也可以选择只在选定的特征点处估计 MV,在那些地方估计精度是高的。这里的任务是确定 $\mathbf{a}$,使得模型 $\mathbf{d}(\mathbf{x};\mathbf{a})$能够很好地近似预估计的 MV $\mathbf{d}_n, n\in\mathcal{N}$。这可以通过最小化下面的匹配误差来实现:

$$E_{\mathrm{fit}} = \sum_{n\in\mathcal{N}} w_n \mid \mathbf{d}(\mathbf{x}_n;\mathbf{a}) - \mathbf{d}_n \mid^p \tag{6.7.2}$$

如 5.5.4 节所说明的,全局运动一般用多项式函数描述或近似。在这种情况下,$\mathbf{a}$ 由多项式的系数组成,$\mathbf{d}(\mathbf{x}_n;\mathbf{a})$是 $\mathbf{a}$ 的线性函数,即 $\mathbf{d}(\mathbf{x}_n;\mathbf{a}) = [\mathbf{A}(\mathbf{x}_n)]\mathbf{a}$。如果选择 $p=2$,那么最小化的问题变成了一个加权的最小平方问题。通过设置$\partial E_{\mathrm{fit}}/\partial\mathbf{a}=0$,我们得到如下的解:

$$\mathbf{a} = \left(\sum_{n\in\mathcal{N}} w_n[\mathbf{A}(\mathbf{x}_n)]^T[\mathbf{A}(\mathbf{x}_n)]\right)^{-1}\left(\sum_{n\in\mathcal{N}} w_n[\mathbf{A}(\mathbf{x}_n)]^T\mathbf{d}_n\right) \tag{6.7.3}$$

例如,考虑在公式(5.5.16)给出的仿射运动模型。运动参数矢量是 $\mathbf{a} = [a_0, a_1, a_2, b_0, b_1, b_2]^T$,矩阵$[\mathbf{A}(\mathbf{x}_n)]$是:

$$[\mathbf{A}(\mathbf{x}_n)] = \begin{bmatrix} 1 & x_n & y_n & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_n & y_n \end{bmatrix}$$

实际上，x 和 y 维的参数并不是成对的，可以分别估计，这就减小了所设计的矩阵的尺寸。例如，估计 x 方向的参数 $\mathbf{a}_x=[a_0,a_1,a_2]$，有关的矩阵是 $[\mathbf{A}_x(\mathbf{x}_n)]=[1,x_n,y_n]$，解为：

$$\mathbf{a}_x=\left(\sum_{n\in\mathcal{N}} w_n[\mathbf{A}_x(\mathbf{x}_n)]^T[\mathbf{A}_x(\mathbf{x}_n)]\right)^{-1}\left(\sum_{n\in\mathcal{N}} w_n[\mathbf{A}_x(\mathbf{x}_n)]^T d_{n,x}\right) \tag{6.7.4}$$

6.8 基于区域的运动估计*

正如前一节所指出的，在一个成像的场景中通常有多种类型的运动，它们对应于与不同的物体有关的运动。在基于区域的运动估计中，我们把图像帧(锚定帧)分割成多个区域，并估计每个区域的运动参数。这种分割应该使一个单一的参数运动模型可以很好地表示每个区域的运动。显然，区域分割依赖于用于表征每个区域的运动模型。最简单的方法是要求每个区域进行单独的平移运动。然而这个要求会造成太多小的区域，因为在对应于一个物理物体的区域中的二维运动，极少能够用一个简单的平移来模型化。这样一个区域必须分割成许多小的子区域，使每一个子区域具有单一的平移运动。对于更高效的运动表示，应该使用仿射、双线性或透视运动模型。

一般实现基于区域的运动估计有 3 种方法。第一种方法，我们首先把图像帧分割成不同的区域——基于纹理同质性、边缘信息以及有时通过对两帧间不同图像的分析得到的运动边界，然后估计每一个区域中的运动。后者可以通过对每个区域应用 6.7 节描述的全局运动估计法实现。我们称这种方法为区域优先。第二种方法，我们首先估计整个图像的运动场，然后分割得到的运动场，使得每一个区域的运动可以用单一的参数模型描述，我们称这种方法为运动优先。得到的区域可以在一些空间的连通性约束下进一步地优化。这个方法中的第一步可以用前面描述的各种运动估计方法实现，包括基于像素、块和网格的方法。第二步涉及基于运动的分割，将在 6.8.1 节中进一步讨论。第三种方法是对区域分割和每一个区域的运动进行联合估计。一般这是用一个迭代过程实现的，交替地进行区域分割和运动估计；在6.8.2节介绍这种方法。

6.8.1 基于运动的区域分割

如前所述，基于运动的分割是指把运动场分成多个区域，使每个区域中的运动都可由一个单一的运动参数集来描述。这里我们给出实现这个任务的两种方法：第一种方法使用聚类技术确定相似的运动矢量。第二种方法用分层技术从占主导运动的区域开始，相继地估计区域和相应的运动。

聚类　考虑每个区域的运动模型是纯平移的情况。分割的任务是把所有具有类似 MV 的空间相连的像素分组到一个区域。这用自动聚类方法，例如 K 平均或 ISODATA 方法[8]很容易实现。这是一个迭代过程：从一个初始分割开始，计算每个区域的平均运动矢量，称为质心。然后每个像素被重新划分到其质心最接近这个像素的运动矢量的区域。这产生了一个新的分割，前面的两步可以循环进行，直到分割不再变化为止。在这个过程中，没有考虑空间的连通性。因此，得到的区域可能包含空间不连通像素。在迭代的末尾可以加一个后处理步骤，以改

* 标有星号的节可以跳过或者留待进一步研究。

进所得区域的空间连通性。例如，一个单一区域可以分成几个子区域，使得每个区域都是一个空间连通的子集，孤立的像素可以合并到它周围的区域中，最后，区域边界可以使用形态学算子进行平滑。

当每个区域的运动模型不是一个简单的平移时，基于运动的聚类不是简单的。这是因为我们不能用 MV 之间的相似性作为准则来进行聚类。一种解决办法是通过用像素邻域内的 MV 匹配一个指定的模型，为每一个像素寻找一个运动参数集。然后我们可以利用前面描述的聚类方法，用运动参数矢量替换原始的 MV。如果原始运动场是用高阶模型的基于块的表示给出的，那么可以把有相似运动参数的块聚类到同一个区域中。类似地，使用基于网格的运动表示，对于每一个基于节点位移的元素，都可以导出一个运动参数集，然后把具有相似参数的元素聚类到同一个区域中。参考文献[44]描述了这种并行的方法。

分层 通常，场景中的运动场可以分解为层，用第一层表示主导的运动，第二层表示次主导的运动，等等。这里，运动的主导性是由进行相应运动的区域范围决定的。主导的运动通常反映摄像机的运动，它影响整个图像域。例如，在网球比赛的视频剪辑中，背景将是第一层，它一般进行一致的全局摄像机运动；运动员是第二层，它通常包含相应于身体不同部位的运动的几个子物体级运动；球拍是第三层；球是第四层。为了提取不同层的运动参数，我们可以递归地使用 6.7.1 节描述的健壮估计方法。首先，尝试用单个参数集来模型化整个帧的运动场，并且连续地从剩余的内围层组去掉外露层像素，直到所有的内围层组中的像素能够被很好地模型化。这将产生第一个主导区域(相应于内围层区域)和与之相关的运动。然后对剩余的像素(外露层区域)应用同样的方法，确定次主导区域及其运动。持续进行这个过程直到没有外露层像素为止。同前面一样，在迭代的末尾可启用后处理以改善所得区域的空间连通性。这就是参考文献[44]中描述的顺序方法。

为了使这种方法能很好地工作，在任何一次迭代中，内围层区域都必须明显大于外露层区域。这意味着最大的区域必须大于所有其他区域的联合，次最大区域必须大于剩余区域的联合，等等。这个条件在大多数视频场景中是满足的，它通常含有一个静止的覆盖大部分图像的背景，以及具有变化尺寸的不同的运动物体。

6.8.2 联合区域分割和运动估计

理论上，我们可以把区域分割图和每个区域运动参数的联合估计公式化为一个最优化问题。最小化目标函数可以是运动补偿预测误差和区域平滑度量的联合。然而，因为很高维的参数空间和这些参数之间复杂的互相依赖关系，解这个最优化的问题是困难的。实际上，经常采用次最优化的方法，它轮换地进行分割估计和运动参数估计。基于初始的分割，首先估计每一个区域的运动。在下一次迭代中，优化这个分割——例如，去掉每个预测误差大的区域中的外露层像素，合并共用相似运动模型的像素。然后重新估计每个优化区域的运动参数。持续这个过程直到分割图不再发生变化为止。

另外一个方法是以分层的方式估计区域及其有关的运动，类似于前面所述的分层的方法。这里我们假定每一个点的运动矢量是已知的，具有最主导运动区域(即内围层)的确定是由检验因使用一个运动参数集表示各个 MV 所造成的匹配误差实现的。这实质上是 6.7.3 节中介绍的间接健壮估计器。在联合区域分割和运动估计方法中，为了从剩余的像素中提取次主导区域和相关的运动，我们可以使用直接健壮估计器。即通过最小化这些像素的预测误差来直

接估计运动参数。参数一旦确定,通过检验这个像素的预测误差,我们就可以确定这个像素是否属于内围层组。然后通过只最小化内围层像素的预测误差,来重新估计运动参数。这种方法已经由 Hsu , Anandan 和 Peleg[18]采取。

6.9　多分辨率运动估计

从前面几节可以看到,各种运动估计的方法可以简化为求解一个误差最小化问题。有两个与得到正确的解有关的主要困难:(1) 最小化函数一般有许多局部最小,而且不容易达到全局最小,除非它很接近所选择的初始解;(2) 最小化过程所涉及的计算量是很大的。用多分辨率的方法可以解决这两个问题,它以依次精细的分辨率搜索最优化问题的解。通过首先搜索粗分辨率的解,通常可以得到一个接近于真实运动的解。此外,通过把每个较细分辨率的搜索限制在前一个分辨率中获得的解的一个小邻域内,与以最细的分辨率在一个大的范围内直接搜索相比,搜索的总数可以降低。

在本节中,我们首先描述一般背景下运动估计的多分辨率方法,它可以应用到任何运动模型。然后我们集中于块平移模型,并描述一个分层的块匹配算法。

6.9.1　一般公式

如图 6.19 所示,首先取得两个原始图像帧的金字塔表示,其中每一级都是下面一级的降低分辨率的表示,它是通过空间低通滤波和亚采样获得的。底级是原始图像。然后估计两个金字塔的相应级间的运动场,从顶级(最粗的)开始,重复地前进到下一个较细的级。在每一个新的较细分辨率级,对在前一个较粗分辨率级得到的运动场进行内插,形成当前级运动的初始解。最常见的金字塔结构是,它的分辨率在相继级之间的水平和垂直方向上都是减半的。通常,用一个简单的 2×2 平均滤波器作为低通滤波器;为了获得更好的性能,可以采用高斯滤波器。

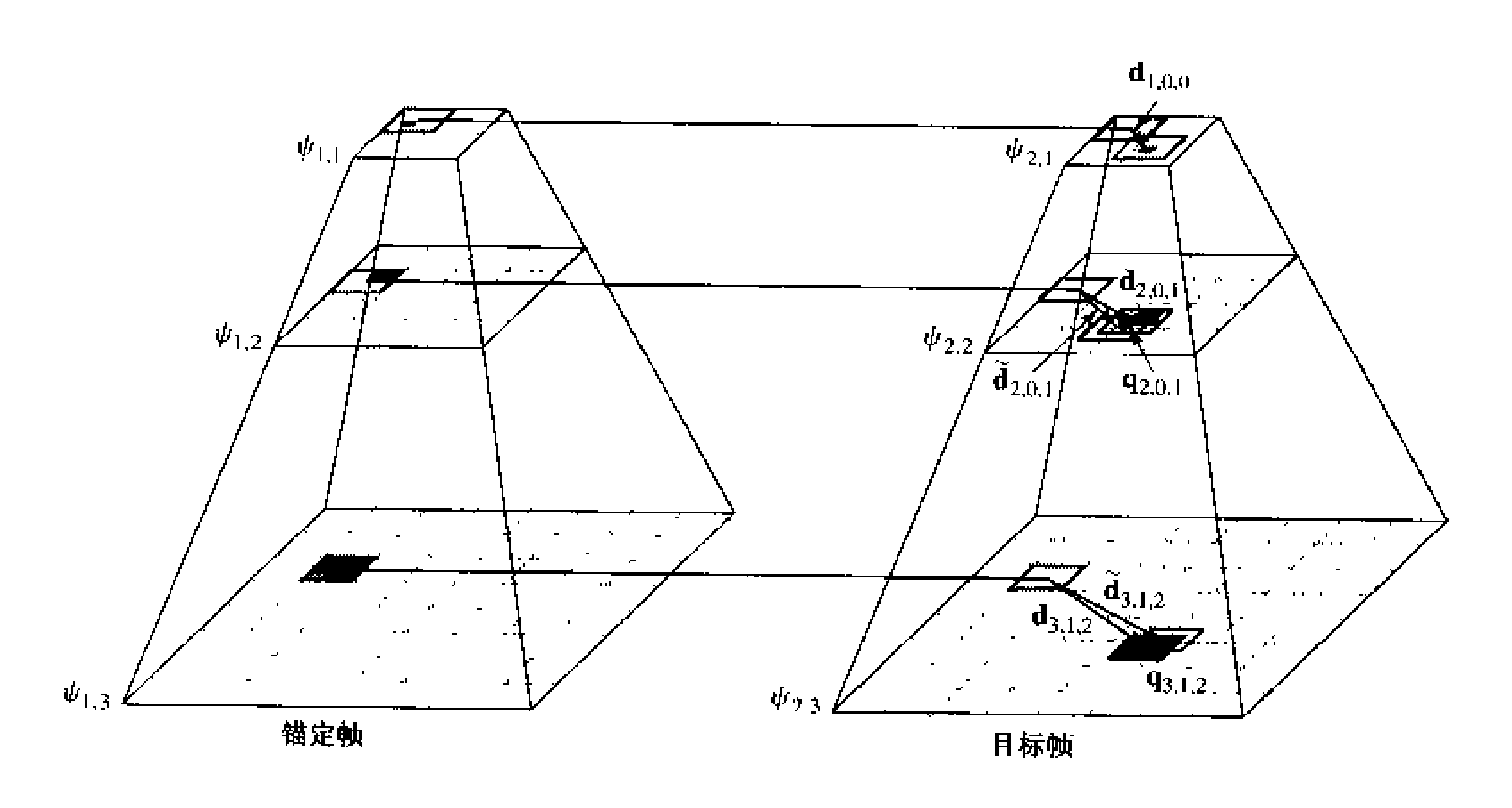


图 6.19　分层的块匹配算法的图示

假定级数是 L,以第 L 级为原始图像。令锚定帧和目标帧的第 l 级图像表示为$\psi_{t,l}(\mathbf{x})$,$\mathbf{x}\in\Lambda_l$,$t=1,2$,其中 Λ_l 是第 l 级的像素集合。用 $\mathbf{d}_{l-1}(\mathbf{x})$表示从第 1 级到第 $l-1$ 级获得的全部的运动场。在第 l 级,我们首先把 $\mathbf{d}_{l-1}(\mathbf{x})$内插到级 l 的分辨率,以产生一个初始运动估计 $\tilde{\mathbf{d}}_l(\mathbf{x})=\mathcal{U}(\mathbf{d}_{l-1}(\mathbf{x}))$,其中$\mathcal{U}$表示内插算子。然后确定这一级的更新 $\mathbf{q}_l(\mathbf{x})$,使误差:

$$\sum_{\mathbf{x}\in\Lambda_l}|\psi_{2,l}(\mathbf{x}+\tilde{\mathbf{d}}_l(\mathbf{x})+\mathbf{q}_l(\mathbf{x}))-\psi_{1,l}(\mathbf{x})|^p \tag{6.9.1}$$

为最小。这一步以后得到的新运动场是 $\mathbf{d}_l(\mathbf{x})=\mathbf{q}_l(\mathbf{x})+\tilde{\mathbf{d}}_l(\mathbf{x})$。当相继的精细化完成以后,在最精细的分辨率上总的运动是:

$$\mathbf{d}(\mathbf{x})=\mathbf{q}_L(\mathbf{x})+\mathcal{U}(\mathbf{q}_{L-1}(\mathbf{x})+\mathcal{U}(\mathbf{q}_{L-2}(\mathbf{x})+\cdots+\mathcal{U}(\mathbf{q}_1(\mathbf{x})+\mathbf{d}_0(\mathbf{x}))\cdots)) \tag{6.9.2}$$

这个程序的初始条件是 $\mathbf{d}_0(\mathbf{x})=0$。我们可以直接指定总运动 $\mathbf{d}(\mathbf{x})$,或者指定在所有级上的运动更新 $\mathbf{q}_l(\mathbf{x})$,$l=1,2,\cdots,L$。后者表示分层结构的运动,在需要渐进地恢复运动场的应用中希望使用这种表示。

多分辨率方法的优点是双重的。首先,最小化的问题在较粗分辨率下比在较细分辨率下状态要好;因此,在较粗级上得到的解更有可能接近该分辨率的真正的解。这个解到下一个分辨率级的内插则提供一个更接近该级真正解的初始解。重复这个相继的从最粗到最细分辨率的步骤,在最细分辨率上得到的解就很可能接近真正的解(全局最小)。第二,在每个分辨率级上的估计可以被限制在比最细分辨率上的真正运动区域小很多的搜索范围内,使得要实施的搜索总数比在最细分辨率上直接搜索所需的搜索数要小。实际的搜索数依赖于设置在不同级的搜索范围。

图像处理的多分辨率表示首先由 Burt 和 Adelson[6]介绍。在运动估计中的应用依赖于使用的运动模型。在前面的表述中,我们假定所有像素的 MV 都是要被估计的。这个算法可以很容易地修改为估计基于块、基于网格、全局或物体级的运动参数。因为块平移运动模型在实际应用中是非常流行的,所以下面我们较详细地考虑这种特殊的情况。

6.9.2 分层块匹配算法

如 6.4.1 节所指出的,使用穷尽搜索法得到块运动矢量(MV)需要极大的计算量。除此以外,估计的块 MV 经常会导致混乱的运动场。在本节中,我们介绍一种分层的块匹配算法(HBMA),它是刚刚介绍的多分辨率方法的一种特殊情况。这里,锚定帧和目标帧都用金字塔表示,EBMA 或它的快速变体之一被用来估计金字塔中每一级的块的 MV。图 6.19 示出了这个过程,此时金字塔每增加一级,空间分辨率在水平和垂直方向减半。这里,我们假定在不同的级上使用同样大小的块,从而每一维上块的数量也要减半。令在级 l 上块(m,n)的 MV 用 $\mathbf{d}_{l,m,n}$表示。从级 1 开始,首先寻找在这一级中所有块的 MV$\mathbf{d}_{1,m,n}$。在每一个新级 $l>1$,对每个块,它的初始 MV $\tilde{\mathbf{d}}_{l,m,n}$是根据下式用级 $l-1$ 中相应的块内插得到的:

$$\tilde{\mathbf{d}}_{l,m,n}=\mathcal{U}(\mathbf{d}_{l-1,\lfloor m/2\rfloor,\lfloor n/2\rfloor})=2\mathbf{d}_{l-1,\lfloor m/2\rfloor,\lfloor n/2\rfloor} \tag{6.9.3}$$

然后搜索到一个校正矢量 $\mathbf{q}_{l,m,n}$,得到最终估计的 MV:

$$\mathbf{d}_{l,m,n} = \tilde{\mathbf{d}}_{l,m,n} + \mathbf{q}_{l,m,n} \tag{6.9.4}$$

例 6.3　在图 6.20 中，我们示出了两个 32×32 的视频帧，图中锚定帧中的灰块运动的位移为 (13,11)。我们说明如何使用三级 HBMA 估计块的运动场。每一级使用的块的大小是 4×4，搜索的步长是一个像素。从第一级开始，对于锚定帧的块(0,0)，找到的 MV 是 $\mathbf{d}_{1,0,0} = \mathbf{d}_1 = (3,3)$。当进入第二级时，对于块(0,1)，MV 被初始赋值为 $\tilde{\mathbf{d}}_{2,0,1} = \mathcal{U}(\mathbf{d}_{1,0,0}) = 2\mathbf{d}_1 = (6,6)$。从这个初始的 MV 开始，找到的校正矢量是 $\mathbf{q}_2 = (1,-1)$。得到最终估计的 MV $\mathbf{d}_{2,0,1} = \mathbf{d}_2 = (7,5)$。最后，在第三级，块(1,2)的 MV 初始赋值为 $\tilde{\mathbf{d}}_{3,1,2} = \mathcal{U}(\mathbf{d}_{2,0,1}) = 2\mathbf{d}_2 = (14,10)$。由校正矢量 $\mathbf{q}_3 = (-1,1)$，最后估计的 MV 为 $\mathbf{d}_{3,1,2} = \mathbf{d}_3 = (13,11)$。

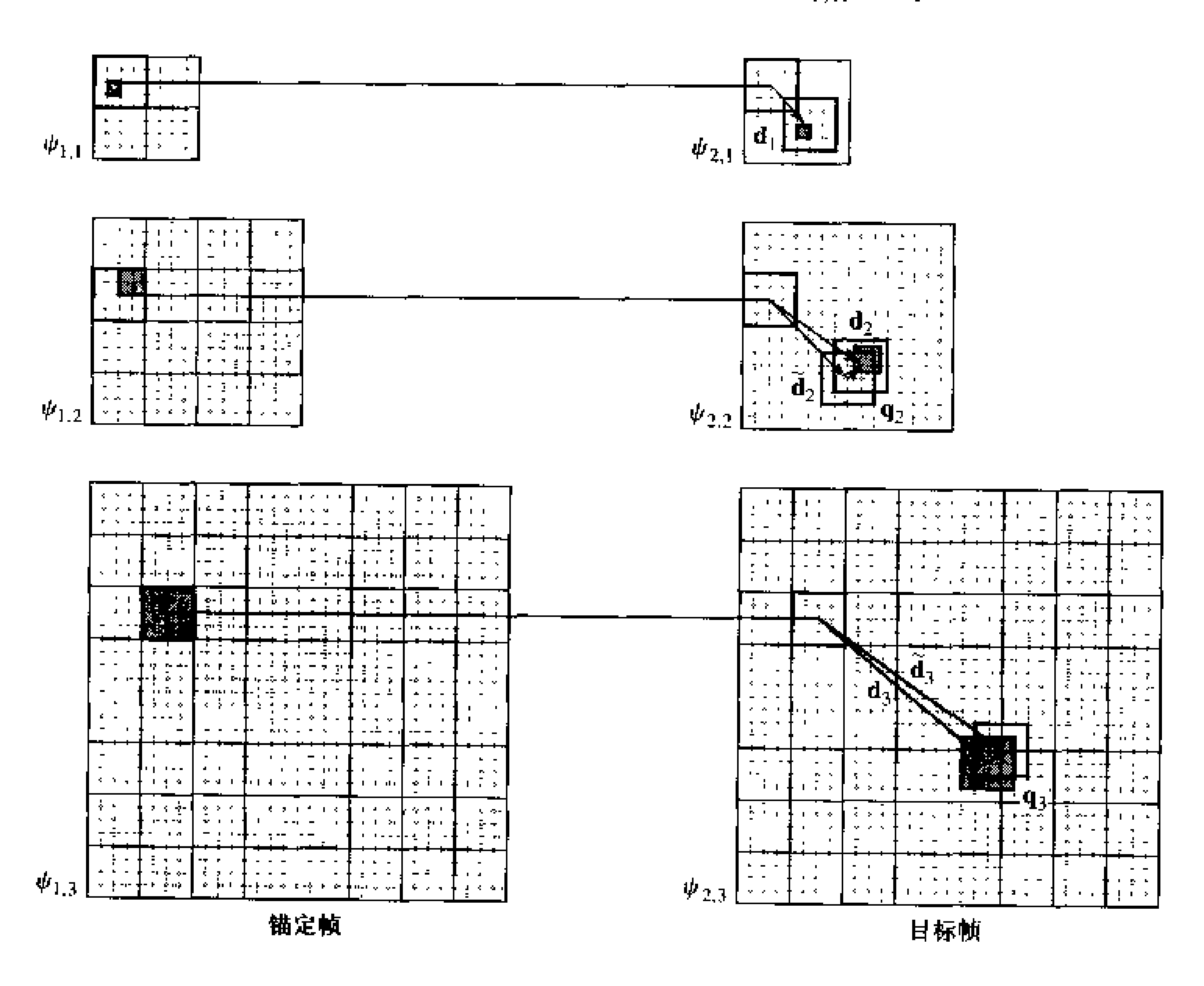


图 6.20　一个三级 HBMA 块运动估计的例子(见例 6.3)

注意，在级 l 使用一个宽度为 N 的块相应于全分辨率下一个宽度为 $2^{L-1}N$ 的块。同样的比例适用于搜索范围和步长。因此，通过在不同的级使用同样的块大小、搜索范围和步长，我们实际上是在搜索开始的时候使用较大的块、搜索范围和步长，然后在以后的步骤中逐渐缩小它们(减半)。

HBMA 中涉及的运算数依赖于每一级的搜索范围。如果在最细分辨率下期望的搜索范围是 R，那么对于一个 L 级的金字塔，我们可以设定第一级的搜索范围是 $R/2^{L-1}$。对于剩下的级，因为由前一级内插得到的初始 MV 通常相当接近真正的 MV，所以校正矢量的搜索范围不需要很大。然而为简单起见，我们假定每一级都使用 $R/2^{L-1}$ 的搜索范围。如果图像的大小是 $M\times M$，每一级块的大小是 $N\times N$，那么第 l 级的块数是 $(M/2^{L-l}N)^2$，搜索的次数是

$(M/2^{L-l}N)^2\cdot(2R/2^{L-1}+1)^2$。因为每一次搜索所需的运算数是 N^2，所以总运算数是：

$$\sum_{l=1}^{L}(M/2^{L-l})^2(2R/2^{L-1}+1)^2=\frac{4}{3}(1-4^{-L})M^2(2R/2^{L-1}+1)^2$$

$$\approx\frac{1}{3}4^{-(L-2)}4M^2R^2$$

回忆 EBMA 的运算量是 $M^2(2R+1)^2\approx 4M^2R^2$（见 6.4.1 节）。因此采用上述参数选择的分层方法将减少计算量 $3\cdot4^{L-2}$ 倍。级 L 的典型数值是 2 或 3。

例 6.4　图 6.21 示出了对于图 6.8 所给出的视频帧，用 HBMA 方法获得的估计结果。在这个例子中，使用了三级金字塔。每一级的搜索范围设置为 4，从而在原始分辨率下的等效搜索范围为 $R=16$。在所有的级都使用整数像素精度搜索。最后的整数精度解在一个像素的搜索范围内用半像素精度搜索进一步优化到半像素精度。将这个最后一级的结果与图 6.8(c) 和 (d) 所示的结果相比较，我们可以看到多分辨率方法确实能够给出比 EBMA 法更加平滑的运动场。视觉观察也可以发现这个运动场更真实地表示了图 6.8(a)

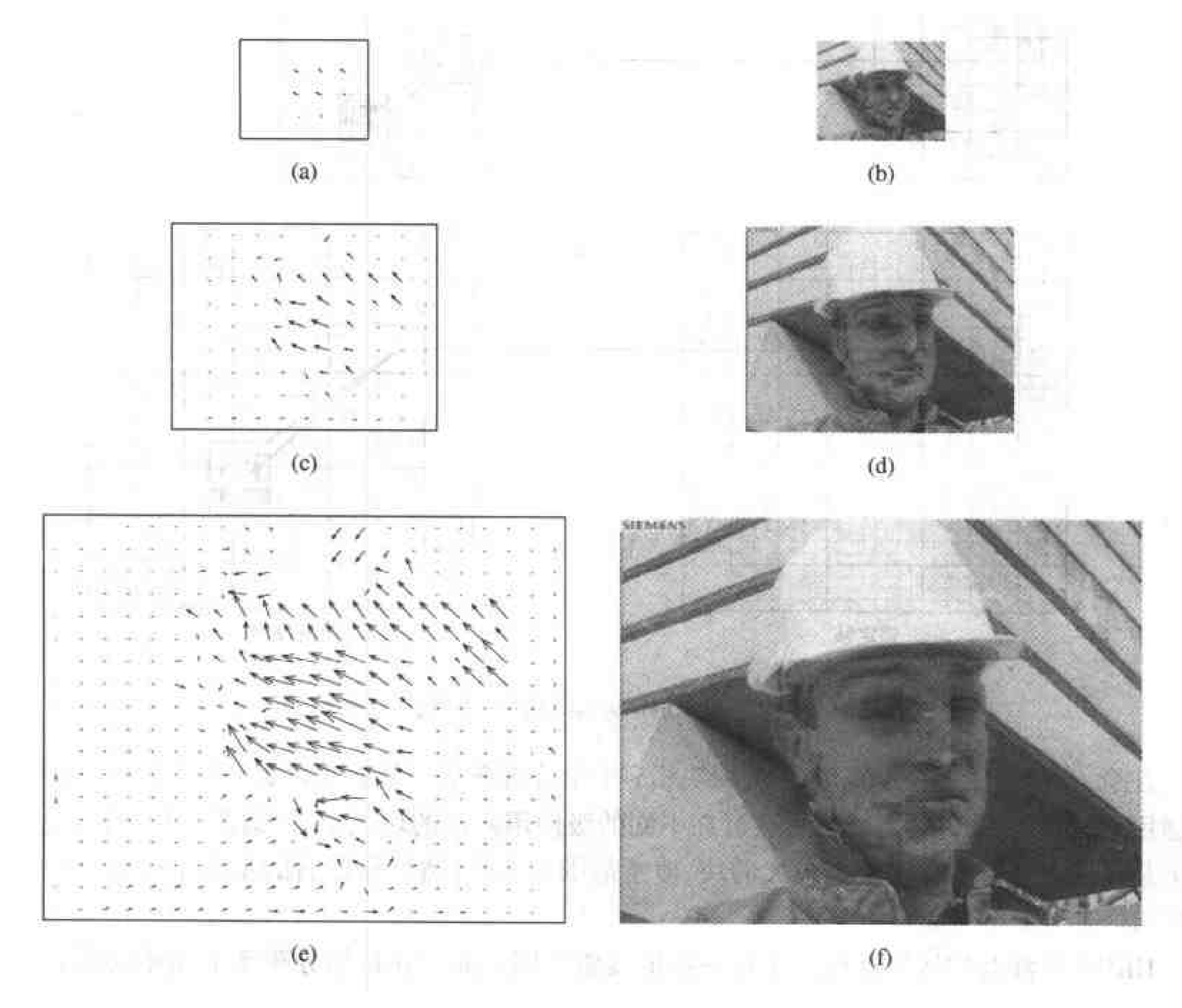


图 6.21　对于图 6.8 所示的两个图像用 HBMA 进行运动估计的例子：(a,b)第一级的运动场和预测图像；(c,d)第二级的运动场和预测图像；(e,f)最后一级的运动场和预测图像（PSNR = 29.32）。这里使用的是三级 HBMA 算法；所有级上的块大小都是 16 × 16。所有级的整数像素精度的搜索范围都是4。最后一级的结果用半像素精度搜索在 ± 1的区域内进一步优化

和(b)的两个图像帧之间的运动。尽管事实是EBMA得到了较高的PSNR,这也是正确的。从计算的复杂性来说,用于图6.8(c,d)的半像素精度EBMA需要$352\times288(4R+1)^2=4.3E+8$次运算,然而如果我们忽略最后的用半像素搜索的再精细化步骤,这里所用的三级算法只需要$\frac{4}{3}\cdot352\times288\left(\frac{R}{2}+1\right)^2=1.1E+7$次运算。

有许多不同的HBMA实现。Bierling首先将这个思想应用于基于块的运动模型[5]。分层BMA的特殊情况被称为可变尺寸或四叉树BMA,它从一个较大的块尺寸开始,然后重复地把一个块分成四块,只要这个块的匹配误差仍比一个阈值大。在这种情况下,所有的处理都是在原始图像的分辨率上进行的。

6.10　运动估计在视频编码中的应用

在任何视频编码系统中运动估计都是一个关键部分。正如将要在9.3.1节讨论的,高效的视频编码方法要使用块时间预测,在这种方法中,将要编码的帧中的块是由前面的编码帧中与它相应的块预测出来的,前面的编码帧称为参考帧①,然后预测误差被编码。为了编码预测误差的比特率最小化,用适当的运动估计准则最小化预测误差。在这类应用中,估计的运动场不必与真实的运动场相像。因此,块匹配算法(EBMA及其包括HBMA的快速变体)可提供简单而有效的解。除了把每个块估计出的MV直接用于该块的预测以外,还可以基于为它的相邻块估计的MV使用这些预测值的加权平均。这被称为重叠块运动补偿,将在9.3.2节中讨论。

注意,在这个视频编码方法中,除了预测误差要编码以外,MV也必须编码。因此,单独最小化预测误差不是要用的最好的准则。由于较平滑的运动场需要较少的编码比特,如果适当地在估计的运动场中施加平滑性,能有助于提高整体的编码效率。因此先进的运动估计算法是通过最小化用于编码MV和预测误差的总比特率工作的。这个课题将在9.3.3节进一步讨论。

为了克服由基于块的运动估计方法产生的块效应,可以应用高阶基于块(DBMA)、基于网格,或者联合应用基于块、基于网格和/或DBMA方法。然而,这些较复杂的方法一般都不会使编码效率有显著的提高。

在较先进的视频编码方法(第10章将介绍)中,通常在基于块的运动估计之前,把全局运动估计(6.7节)用于整个帧,以补偿摄像机运动的影响。而且,整个帧通常分割成几个区域或物体,用6.8节介绍的基于区域的运动估计法估计每一个区域或物体的运动参数。

6.11　小结

图像亮度与运动之间的关系(6.1节)

- 几乎所有的运动估计算法都基于恒定亮度假定(公式6.1.1或5.2.11),或者基于这个假设导出的光流方程(公式6.1.3)。这使我们能够依据一些运动模型,通过确定具有相似亮度的像素估计运动。注意,这种假设仅当照明源是环境的和时不变的,而且物体表

① 见6.2节的脚注。

面是漫反射的情况下才成立(5.2节)。

- 当运动方向垂直于图像亮度的梯度时,或者图像梯度为零时,运动不会引起图像亮度变化。这是基于亮度的运动估计方法的固有限制。

运动估计中的关键元素

- *运动表示*(6.2.1节):这依赖于用来划分帧的分割方法(基于像素、基于块、基于网格、基于区域、全局)、每一个分割区域所用的运动模型(块、网格元素、物体区域、或整个帧)和在邻近区域运动之间的约束。不同的运动表示导致不同的运动估计方法,包括基于像素(6.3节)、基于块(6.4~6.5节)、基于网格(6.6节)、基于区域(6.8节)和全局运动估计(6.7节)。
- *运动估计准则*(6.2.2节):我们给出了估计每个区域上的运动参数的三个准则:(1)最小化DFD误差(当运动较小时,这等价于基于光流方程的方法);(2)当最小化DFD误差时,使产生的运动场穿越区域时尽可能平滑;(3)对于给定的观测帧,最大化运动场的后验概率。我们说明了(3)实质上需要(1)和(2)。除了最小化DFD以外,当区域中的运动是纯平移时,我们也可以检测相位相关函数中的峰值。
- *最优化方法*(6.2.3节):对于选定的表示和准则,运动估计问题通常转化为一个最优化(最小化或最大化)问题,它可以用穷尽搜索或基于梯度的搜索来实现。为了加速搜索和避免陷入到局部最小,可以使用多分辨率方法(6.9节)。

6.12 习题

6.1 叙述不同运动表示方法(基于像素、基于块、基于网格、基于区域和全局)的优点和缺点。

6.2 叙述穷尽搜索和梯度下降法的优点和缺点。同时,比较一阶和二阶梯度下降法。

6.3 与单分辨率方法相比,多分辨率估计方法的主要优点是什么?它有什么缺点吗?

6.4 在6.3.2节,我们用梯度下降法导出了多点邻域法。用光流方程能找到一个封闭形式的解吗?在什么条件下这个解是成立的?

6.5 在6.4.1节中,我们描述了用基于块的运动表示来确定块MV的一个穷尽搜索算法。用光流方程能找到一个封闭形式的解吗?在什么条件下这个解是成立的?

6.6 在公式(6.2.7)中,我们说明了如果运动场是一个常量,可以用光流方程建立一个最小平方问题,并获得一个封闭形式的解。假定运动场不是一个常量,但是能用多项式映射来模型化。能为多项式系数找到一个封闭形式的解吗?(提示:任何多项式映射函数都能表示成 $\mathbf{d}(\mathbf{x};\mathbf{a}) = [\mathbf{A}(\mathbf{x})]\mathbf{a}$,其中 $\mathbf{a}$ 包含所有的多项式系数。)

6.7 在6.4.5节我们说过,当在 $\phi_1(\mathbf{x})$ 的区域块中有几个进行不同运动的小块时,在PCF中将会有几个峰值。每一个峰值对应于一个小块的运动。峰值的位置指示小块的MV,而峰值的幅度与小块的尺寸成比例。请至少定性地证明这个陈述。为简化推导,可以只考虑一维的情况。

6.8 对于EBMA,计算需求依赖于块的大小吗?

6.9 在6.9.2节中,我们导出当每一级的搜索范围是 $R/2^{L-l}$ 时,HBMA所需要的运算次

数。如果除了第一级搜索范围设置为 $R/2^{L-1}$以外,我们在每一级所用的搜索范围是 ±1 个像素,所需要的运算次数是多少呢？这种参数建立得是否适当?

6.10 考虑一个 Y 分量帧的大小为 720×480 的 BT.601 格式视频。比较一个块尺寸为 16×16的 EBMA 算法(整像素)与两级 HBMA 算法所需要的计算量。假定最大运动范围是 ±32。可以用运算数来比较计算量,每一次运算包括一个减法、一次绝对值计算和一次加法。你可以自己假设 HBMA 不同级上的搜索范围。为了简单起见,忽略产生金字塔所需的计算,并假定只进行整数像素搜索。

6.11 对于三级 HBMA 算法,重复习题 6.10。

6.12 写出用整数像素精度实现 EBMA 的 C 或者 MATLAB 代码。用 16×16 的块尺寸。程序应该允许用户选择搜索范围,因此可以比较用不同搜索范围所得的结果(注意,合适的搜索范围依赖于试验图像中的运动范围)。把这个程序应用于视频序列的两个相邻的帧。程序应该产生并绘制估计的运动场、预测的图像和预测误差的图像。它也应该计算预测帧与原始锚定帧相比的 PSNR。对于 MATLAB,可以使用函数 quiver 绘制运动场。

6.13 对于半像素精度的 EBMA 重复习题 6.12。比较用整数像素精度与用半像素精度获得的预测图像的 PSNR。哪一种方法有更精确的预测？哪一种方法需要更多的计算时间?

6.14 由基于块的运动场的空间内插,可以得到高密度的(即基于像素的)运动场。写出能够对习题 6.12 得到的运动场进行内插的 C 或者 MATLAB 代码,假定每个块的 MV 实际上是这个块中心的 MV。使用双线性内插。用内插的像素运动场,可以再一次得到预测图像和预测误差图像。将在习题 6.12 和6.13中得到的运动场、预测图像和预测误差图像与这里得到的进行比较。哪一种方法有更精确的预测？哪一种方法需要更多的计算时间?

6.15 用 C 或者 MATLAB 代码实现 HBMA 算法。可以选择使用两级或者三级分辨率。可以在所有级上使用整数像素搜索,但是在 ±1 邻域内用半像素精度搜索优化最后的结果。所有级上使用的块的尺寸都是 16×16。搜索范围的选择应该使等效的原始分辨率搜索范围为 ±32。在精度和计算时间方面,将这个结果与在习题6.12和6.13 中得到的结果进行比较。

6.16 在 6.7 节,我们说公式(6.7.2)中的匹配误差是用公式(6.7.3)中得到的解最小化的。证明这个结果。

6.17 假定两帧之间的运动可以用一个全局仿射模型来模型化。我们想基于 DFD 准则直接估计仿射参数。建立这个最优化问题,并且导出一个解这个最优化问题的迭代算法。可以使用一阶梯度下降法或者牛顿-拉夫森方法。写出实现你的程序的 C 或者 MATLAB 代码。把它应用在进行摄像机运动的两个视频帧上。将产生的运动场和预测帧与用 EBMA 得到的结果进行比较。

6.18 重复习题 6.17,但是使用间接方法从给定的块 MV 导出仿射参数。导出回归方程和封闭形式的解。写出实现你的算法的 C 或者 MATLAB 代码。可以使用前面为整数像素 EBMA 编写的代码产生块 MV。把这里获得的结果与直接方法获得的结果进行比较(习题 6.17)。

6.13 文献目录

[1] Aggarwal, J. K., and N. Nandhahumar. On the computation of motion from sequences of images--a review. *Proceedings of the IEEE* (1988), 76:917–35.

[2] Altunbasak, Y., and A. M. Tekalp. Closed-form connectivity-preserving solutions for motion compensation using 2-D meshes. *IEEE Trans. Image Process.* (Sept. 1997), 6:1255–69.

[3] Altunbasak, Y., and A. M. Tekalp. Occlusion-adaptive, content-based mesh design and forward tracking. *IEEE Trans. Image Process.* (Sept. 1997), 6:1270–80.

[4] Barron, J. L., D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision* (1994), 12:43–77.

[5] Bierling, M., Displacement estimation by hierarchical block matching. In *SPIE Conf.: Visual Commun. Image Processing* (Nov. 1988), SPIE-1001:942–51.

[6] Burt, P. J., and E. H. Adelson. The Laplacian pyramid as a compact image code. *IEEE Trans. Commun.*(1983), COM-31:532–40.

[7] Chou, P., and C. Brown. The theory and practice of bayesian image labeling. *International Journal of Computer Vision* (1990), 4:185–210.

[8] Duda, R. O., and P. E. Hart. *Pattern Classification and Scene Analysis.* New York: John Wiley & Sons, 1973.

[9] Fleet, D. J., and A. D. Jepson. Computation of component image velocity from local phase information. *International Journal of Computer Vision* (1990), 5:77–104.

[10] Fleet, D. J. Disparity from local weighted phase-correlation. In *IEEE International Conference on Systems, Man, and Cybernetics: Humans, Information and Technology* (1994), 48–54.

[11] Geman, S., and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.* (Nov. 1984), 6:721–41.

[12] Girod, B. Motion compensation: Visual aspects, accuracy, and fundamental limits. In M. I. Sezan and R. L. Lagendijk, eds., *Motion Analysis and Image Sequence Processing.* Boston: Kluwer Academic Publishers, 1993, 126–52.

[13] Girod, B. Motion-compensating prediction with fractional-pel accuracy. *IEEE Transactions on Communications* (April 1993), 41(4):604–12.

[14] Hang, H.-M., Y.-M. Chou, and S.-C. Cheng. Motion estimation for video coding standards. *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology* (Nov. 1997), 17(2/3):113–36.

[15] Haralick, R. M., and J. S. Lee. The facet approach to optical flow. In *Image Understanding Workshop* (1993).

[16] Horn, B. K. P., and B. G. Schunck. Determining optical flow. *Artificial intelligence* (1981), 17(1–3):185–203.

[17] Horn, B. K. P. *Robot Vision.* Cambridge, MA: MIT Press, 1986.

[18] Hsu, S., P. Anandan, and S. Peleg. Accurate computation of optical flow using layered motion representations. In *IEEE Int. Conf. Patt. Recog.* (Oct. 1994), 743–46, Jerusalem, Israel.

[19] Jain, J. R., and A. K. Jain. Displacement measurement and its application in interframe image coding. *IEEE Trans. Commun.* (Dec. 1981), COM-29:1799–808.

[20] Koga, T., et al. Motion-compensated interframe coding for video conferencing. In *Nat. Telecommun. Conf.* (Nov. 1981), G5.3.1–5, New Orleans, LA.

[21] Komarek, T., and P. Pirsch. Array architecture for block matching algorithms. *IEEE Trans. Circuits and Systems* (Oct. 1989), 36:269–77.

[22] Konrad, J., and E. Dubois. Bayesian estimation of motion vector fields. *IEEE Trans. Pattern Anal. Machine Intell.* (Sept. 1992), 14:910–27.

[23] Kuglin, C., and D. Hines. The phase correlation image alignment method. In *IEEE Int. Conf. Cybern. Soc.* (1975), 163–65.

[24] Lee, O., and Y. Wang. Motion compensated prediction using nodal based deformable block matching. *Journal of Visual Communications and Image Representation* (March 1995), 6:26–34.

[25] Lee, X., and Y.-Q. Zhang. A fast hierarchical motion compensation scheme for video coding using block feature matching. *IEEE Trans. Circuits Syst. for Video Technology* (Dec. 1996), 6:627–35.

[26] Mitiche, A., Y. F. Wang, and J. K. Aggarwal. Experiments in computing optical flow with gradient-based, multiconstraint method. *Pattern Recognition* (June 1987), 20(2):173–79.

[27] Musmann, H. G., M. Hotter, and J. Ostermann. Object oriented analysis-synthesis coding of moving images. *Signal Processing: Image Commun.* (Oct. 1989), 1:119–38.

[28] Musmann, H. G., P. Pirsch, and H.-J. Grallert. Advances in picture coding. *Proceedings of the IEEE* (April 1985), 73(4):523–48.

[29] Nagel, H. H. Displacement vectors derived from second-order intensity variations in images sequences. *Computer Graphics and Image Processing* (1983), 21:85–117.

[30] Nagel, H. H., and W. Enklemann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. Pattern Anal. Machine Intell.* (Sept. 1986), 8:565–93.

[31] Netravali, A. N., and J. D. Robbins. Motion-compensated coding: some new results. *Bell System Technical Journal* (Nov. 1980), 59(9):1735–45.

[32] Pirsch, P., N. Demassieux, and W. Gehrke. VLSI architecture for video compression—a survey. *IEEE* (Feb. 1995), 83:220–46.

[33] Reddy, B. S., and B. N. Chatterji. An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE Trans. Image Process.* (Aug. 1996), 5(8):1266–71.

[34] Barron, A., J. Rissanen, and Y. Bin. The minimum description length principle in coding and modeling. *IEEE Trans. Information Theory* (Oct. 1998), 44(6):2743–60.

[35] Rousseeuw, P. J., and A. M. Leroy. *Robust Regression and Outlier Detection.* New York: John Wiley & Sons, 1987.

[36] Seferidis, V., and M. Ghanbari. General approach to block matching motion estimation. *Optical Engineering* (July 1993), 32:1464–74.

[37] Shekarforoush, H., M. Berthod, and J. Zerubia. Subpixel image registration by estimating the polyphase decomposition of cross power spectrum. In *IEEE Com-*

puter Society Conference on Computer Vision and Pattern Recognition (1996), 532–37.

[38] Stiller, C., and J. Konrad. Estimating motion in image sequences. *IEEE Signal Processing Magazine* (July 1999), 16:70–91.

[39] Tekalp, A. M., *Digital Video Processing*. Upper Saddle River, NJ: Prentice Hall PTR, 1995.

[40] Thomas, G. A. Television motion measurements for DATV and other applications. Research report 1987/11 (September), British Broadcasting Corporation.

[41] Wang, Y., and O. Lee. Use of 2-D deformable mesh structures for video compression. Part I—the synthesis problem: Mesh based function approximation and mapping. *IEEE Trans. Circuits Syst. for Video Technology* (Dec. 1996), 6:636–46.

[42] Wang, Y., and O. Lee. Active mesh—a feature seeking and tracking image sequence representation scheme. *IEEE Trans. Image Process.* (Sept. 1994), 3:610–24.

[43] Wang, Y., and J. Ostermann. Evaluation of mesh-based motion estimation in H.263 like coders. *IEEE Trans. Circuits Syst. for Video Technology* (June 1998), 8:243–52.

[44] Wang, Y., X.-M. Hsieh, J.-H. Hu, and O. Lee. Region segmentation based on active mesh representation of motion: comparison of parallel and sequential approaches. In *IEEE International Conference on Image Processing (ICIP'95)* (Oct. 1995), 185–88, Washington, DC.

[45] Zienkewicz, O. C., and R. L. Taylor. *The Finite Element Method*, Vol. 1. 4th ed. Upper Saddle River, NJ: Prentice Hall, 1989.

第7章　三维运动估计

三维运动估计允许描述一个三维空间物体的运动——这个运动由运动物体的图像进行估计。它在计算机视觉应用中用于物体跟踪(如在车辆制导和机器人技术中),以及在基于物体的视频编码中用于物体跟踪和运动补偿。在一些应用中,是运动的摄像机穿过静止的场景;在另一些应用中,则是静止的摄像机观测运动的物体。对于一些例如飞机监测的应用中,摄像机和物体都是运动的。

与二维运动估计一样,三维运动估计是一个病态的问题。为了保证得到有用的解,我们做出一些简化的假设——例如刚性物体,通常是已知形状的。任务是估计公式(5.3.9)中给出的物体的六个运动参数。如果物体的形状是未知的,大多数算法假设物体的形状是平面或是抛物线的。

对于三维运动估计,可以用正交或者透视投影作为摄像机模型。显然使用正交投影可简化估计过程。然而,正交投影不允许我们观测深度,并且如果物体沿着摄像机光轴运动或者是绕着一个不是光轴的轴转动,就会产生估计误差。

另一种三维运动估计算法的分类包括直接方法(用图像信号作为其输入)和间接方法(依赖于用一个分离算法(例如块匹配或特征匹配)建立的特征对应)。在7.1节中,我们将讨论用特征对应(feature correspondence)估计运动的算法。估计的运动参数的精度在很大程度上依赖于特征对应的精度。给定这些对应,这个算法就可以很容易地估计大的运动了。由于运动不是直接由图像信号,而是由从图像信号得到的特征估计出来的,我们称这种算法为间接运动估计方法。在7.2节中,我们给出直接运动估计方法。这种方法的优点是不依赖于特征的对应。然而,这种算法只能估计小的运动。因此,经常用在一个迭代估计循环中。

7.1　基于特征的运动估计

基于特征的估计假设一个物体上的同样的特征已经在两个图像中被识别出来。为了允许可靠的特征检测,这些特征通常相应于必须估计运动的物体上的物理标志。特征可以被描述为图像平面中的点、线或角(线之间的连接)。在图像平面中,位移矢量可以用来建立特征点之间的对应。特征的对应可以用简单的块匹配确定。如果把匹配限制于具有高空间梯度的图像区域中,就可以得到比较可靠的结果。因此我们首先提取两个图像的边缘,然后尝试匹配这些边缘。本节将把我们的讨论限制在依赖于特征对应的算法上。对于线和角对应的讨论,见参考文献[2]或参考文献[11]中的综述。

在后面的几节中,我们将讨论基于特征运动估计的4个算法。每个算法都假设针对的是刚性的三维物体。第一个算法(7.1.1节)假设具有正交投影的摄像机模型和已知物体形状。第二个算法(7.1.2节)假设具有透视投影的摄像机模型和已知物体形状。第三个算法(7.1.3节)适合于能用平面近似的物体。最后,我们给出一个基于透视投影和不需要任何关于物体形状知识的算法(7.1.4节)。

7.1.1 正交投影下已知形状的物体

在本节中,我们假设知道运动物体的形状。因此对于图像 k 中的每一个点 $\mathbf{x}=[x,y]^T$,我们知道与 $\mathbf{x}$ 相联系的物体表面点的三维坐标 $\mathbf{X}=[X,Y,Z]^T$。根据图 5.2 (b)和公式(5.1.3),我们采用一个正交投影的摄像机模型。使用这个摄像机模型和三维运动的公式(5.3.9),运动前后点 $\mathbf{X}$ 到图像平面的映射变成:

$$\mathbf{x}'=\begin{pmatrix}x'\\y'\end{pmatrix}=\begin{pmatrix}r_1x+r_2y+(r_3Z+T_x)\\r_4x+r_5y+(r_6Z+T_y)\end{pmatrix} \tag{7.1.1}$$

具有旋转矩阵:

$$[\mathbf{R}]=\begin{bmatrix}r_1 & r_2 & r_3\\r_4 & r_5 & r_6\\r_7 & r_8 & r_9\end{bmatrix} \tag{7.1.2}$$

根据公式(5.3.6)和平移矢量 $\mathbf{T}=[T_x,T_y,T_z]^T$。这是图像 k 中的点 $\mathbf{x}$ 到图像 $k+1$ 中的点 $\mathbf{x}'$ 的一个仿射映射。按照公式(5.3.6)~(5.3.10),线性化旋转矩阵的元素 r_i,我们得到 $\mathbf{x}$ 与 $\mathbf{x}'$ 之间的仿射关系:

$$\begin{pmatrix}x'\\y'\end{pmatrix}=\begin{pmatrix}x-\theta_zy+(\theta_yZ+T_x)\\\theta_zx+y-(\theta_xZ+T_y)\end{pmatrix} \tag{7.1.3}$$

给定至少三点的对应,使用最小平方法,5 个未知运动参数 $\theta_x,\theta_y,\theta_z$ 和 T_x,T_y 的解可以估计出来。为了提高估计运动参数的可靠性,建议最少使用 6 个对应。

公式(7.1.3)使用线性化以后的旋转矩阵。因此估计的运动参数只能近似于真实的运动参数,而且为了估计大的运动参数,算法必须采用迭代的形式。每一次迭代,估计的运动参数都会减少;由于使用线性化的旋转矩阵,误差也会减小。一旦运动参数的更新值为零,迭代就应该停止了。在 7.3 节中,我们给出一个更具体的迭代运动估计的例子。

在公式(7.1.3)中,运动参数 θ_x 和 θ_y 被乘以摄像机中心到物体表面点 $\mathbf{X}$ 之间的距离 Z。因此,只有当物体的形状和 Z 是精确已知的,才可以精确估计 θ_x 和 θ_y。由于物体的形状(或它的深度)并不总是已知的,已经提出了几个估计物体形状连同其运动估计方法的建议。在参考文献[1]中,提出了每次迭代的两步方法,首先估计运动参数,然后更新物体的深度估计。在参考文献[4]中,提出了进一步的改进,根据误差函数的梯度更新深度估计 Z。如果不能利用合理的深度估计,对于主要平行于图像平面平移和绕摄像机轴转动的物体,可以使用公式(7.1.3)。也就是说,我们假设 $\theta_x=\theta_y=0$,并且只估计 θ_z,T_x 和 T_y。

7.1.2 透视投影下已知形状的物体

为了对已知形状的三维物体的任意运动进行估计,我们假设一个已知点 $\mathbf{X}$ 运动到未知的位置 $\mathbf{X}'$。使用公式(5.1.2)的透视投影的摄像机模型把 $\mathbf{X}'$ 投影到图像平面上得到:

$$\mathbf{x}' = \begin{bmatrix} x' \\ y' \end{bmatrix} = F \cdot \frac{1}{Z'} \begin{bmatrix} X' \\ Y' \end{bmatrix} \tag{7.1.4}$$

如果现在用三维运动方程(公式(5.3.9))把 $\mathbf{X}'$替换为 $\mathbf{X}$,我们得到:

$$\mathbf{x}' = F \cdot \begin{bmatrix} \dfrac{r_1 X + r_2 Y + r_3 Z + T_x}{r_7 X + r_8 Y + r_9 Z + T_z} \\ \dfrac{r_4 X + r_5 Y + r_6 Z + T_y}{r_7 X + r_8 Y + r_9 Z + T_z} \end{bmatrix} \tag{7.1.5}$$

问题是求解运动参数 $\theta_x, \theta_y, \theta_z, T_x, T_y, T_z$,假设 $\mathbf{x}, \mathbf{x}'$和 $\mathbf{X}$ 是已知的。为了用线性方程组求解这个方程,根据公式(5.3.10),我们假设小的转动角度并使用线性旋转矩阵。这就得到:

$$\mathbf{x}' = F \cdot \begin{bmatrix} \dfrac{X - \theta_z Y + \theta_y Z + T_x}{-\theta_y X + \theta_x Y + Z + T_z} \\ \dfrac{\theta_z X + Y - \theta_x Z + T_y}{-\theta_y X + \theta_x Y + Z + T_z} \end{bmatrix} \tag{7.1.6}$$

进一步计算以后,我们得到:

$$\begin{bmatrix} x' \cdot \frac{Z}{F} - X \\ y' \cdot \frac{Z}{F} - Y \end{bmatrix} = \begin{bmatrix} -Y\frac{x'}{F} & Z + X\frac{x'}{F} & -Y & 1 & 0 & -\frac{x'}{F} \\ -Z - Y\frac{y'}{F} & X\frac{y'}{F} & X & 0 & 1 & -\frac{y'}{F} \end{bmatrix} \cdot \begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \\ T_x \\ T_y \\ T_z \end{bmatrix} \tag{7.1.7}$$

其中点 $\mathbf{x}'$和 $\mathbf{X}$ 是已知的,而 6 个运动参数是未知的。与前面一样,每个点的对应提供两个方程。为了允许特征点估计的不准确性,应该对三个以上的点用最小平方算法解方程(7.1.7)。由于使用了线性旋转矩阵,我们必须迭代地估计运动参数。

7.1.3 平面物体

估计在空间做任意运动的平面的运动参数是一个重要的问题。通常我们用一个或者更多的平面来近似物体的表面。例如,如果用线框近似物体表面,那么线框上的每一个三角形表示一小块平面。这个算法不假设空间平面方位的任何知识。因此,我们将估计确定平面方位和运动的 8 个参数。

假设摄像机模型具有按照公式(5.1.2)的透视投影,按照公式(5.3.9)做刚性物体运动,并且设物体平面为:

$$aX + bY + cZ = 1 \tag{7.1.8}$$

我们就可以把点 $\mathbf{x}$ 从图像 k 到图像 $k+1$ 的映射描述为如公式(5.5.14)所给出的投影映射[17,5]:

$$[x', y']^T = \left[\frac{a_0 + a_1 x + a_2 y}{1 + c_1 x + c_2 y}, \frac{b_0 + b_1 x + b_2 y}{1 + c_1 x + c_2 y} \right]^T \tag{7.1.9}$$

具有 8 个未知的运动和结构参数 a_i, b_i, c_i(有时称为纯参数[18])。

用至少四个点的对应,我们就可以解这些纯参数. 如果我们限制于四个点的对应,就不允许四个点中有三个点是共线的[6]。为了增加结果的可靠性,我们建议使用 8 个以上的对应,并且使用最小平方法解这些纯参数。

在参考文献[20]中,Tsai 和 Huang 使用奇异值分解(SVD)由 8 个纯参数估计三维运动参数和物体平面的方位。只有纯参数能够可靠地估计出来,这种方法才是有用的。请注意这种估计纯参数的算法不容易用于物体跟踪。由两幅图像进行参数估计以后,物体的方位就固定了。因此,下一幅图像就只有 6 个参数需要估计了。如果我们像参考文献[5]那样仍然估计 8 个纯参数,那么就允许对于一个图像帧的图像平面有两个不同的方位。通常,这个算法只用于后向运动估计;也就是说,对帧 $k+1$ 中的任何一个小的区域,我们假设它对应于三维的一小块平面,但是具有未知的平面方位,并估计它从帧 $k+1$ 到帧 k 的运动以及平面的方位。这不允许我们在多个帧上连续地跟踪同一个三维小块。

7.1.4 使用外极线的未知形状的物体

本节描述一种运动估计算法,它允许我们无需任何物体形状的知识即可估计三维运动和形状[8]。不失一般性,我们把摄像机焦距 F 设定为单位长度。我们假设一个刚性物体和透视投影。从公式(7.1.5)开始,把 $\mathbf{X}$ 替换为它在图像平面上的投影 $\mathbf{x}$,且 $F=1$,我们得到:

$$\mathbf{x}' = \begin{pmatrix} \dfrac{r_1 x + r_2 y + r_3 + T_x/Z}{r_7 x + r_8 y + r_9 + T_z/Z} \\ \dfrac{r_4 x + r_5 y + r_6 + T_y/Z}{r_7 x + r_8 y + r_9 + T_z/Z} \end{pmatrix} \tag{7.1.10}$$

注意这个公式与公式(5.5.13)当 $F=1$ 时是相同的。如果把 $\mathbf{T}$ 和 $\mathbf{Z}$ 乘以一个常量,公式不会改变。因此我们将只能确定 $\mathbf{T}$ 的方向,但不能确定它的绝对长度。它的长度决定于物体的大小以及物体到摄像机的距离。

在 7.1.2 节中,假设物体的形状已知,我们用迭代法和线性化求解了这个非线性方程。使用一个中间矩阵可以无需知道形状而估计运动[8,19]。这个中间矩阵称为 E 矩阵或者本质矩阵,具有 9 个本质参数。在公式(7.1.10)中消去 $\mathbf{Z}$,我们可以确定 $\mathbf{x}$ 与 $\mathbf{x}'$之间的关系为:

$$[x', y', 1]\cdot[\mathbf{E}]\cdot[x, y, 1]^T = 0 \tag{7.1.11}$$

其中

$$[\mathbf{E}] = \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix} = [\mathbf{T}][\mathbf{R}] \tag{7.1.12}$$

矩阵$[\mathbf{T}]$定义为:

$$[\mathbf{T}] = \begin{bmatrix} 0 & T_z & -T_y \\ -T_z & 0 & T_x \\ T_y & -T_x & 0 \end{bmatrix} \tag{7.1.13}$$

旋转矩阵[**R**]是根据公式(5.3.6)定义的。用 $Z \cdot Z'$ 乘公式(7.1.11),我们得到:

$$[X', Y', Z'] \cdot [\mathbf{E}] \cdot [X, Y, Z]^T = 0 \qquad (7.1.14)$$

公式(7.1.14)只用于 $\mathbf{T} \neq 0$ 的运动估计;也就是说,如果物体只是做旋转运动,那么我们不能估计它的运动。因此我们假设感兴趣的物体也是做平移运动的。

外极线　公式(7.1.11)定义了一个相应的图像点 $\mathbf{x}$ 与 $\mathbf{x}'$ 之间的线性依赖关系。从而点 $\mathbf{x}$ 运动以后的位置 $\mathbf{x}'$ 位于一条直线上。这条直线称为外极线,并且它由运动参数按下式定义:

$$a(\mathbf{x}, [\mathbf{E}])x' + b(\mathbf{x}, [\mathbf{E}])y' + c(\mathbf{x}, [\mathbf{E}]) = 0 \qquad (7.1.15)$$

其中

$$\begin{aligned} a(\mathbf{x}, [\mathbf{E}]) &= e_1 x + e_2 y + e_3 \\ b(\mathbf{x}, [\mathbf{E}]) &= e_4 x + e_5 y + e_6 \\ c(\mathbf{x}, [\mathbf{E}]) &= e_7 x + e_8 y + e_9 \end{aligned} \qquad (7.1.16)$$

图7.1示出了一个具有转动[**R**]和平移 **T** 的点 **x** 的外极线。

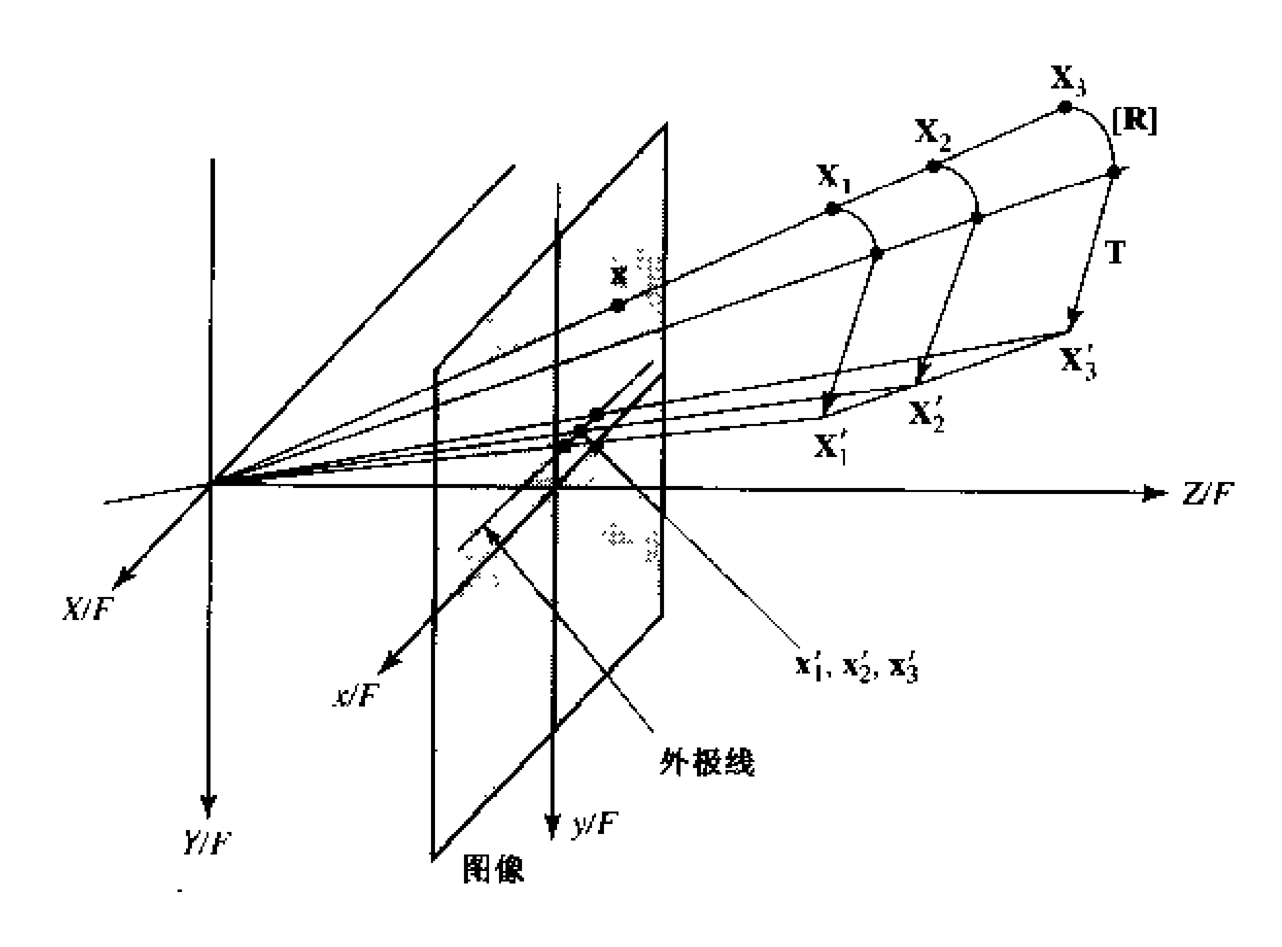


图7.1　用物体运动定义的点 x 的外极线

运动估计用两步执行。首先估计 E 矩阵,然后把它分解成旋转矩阵和平移矢量。

E 矩阵估计　显然当 $\mathbf{T} \neq 0$ 时 E 矩阵不会为零。用公式(7.1.11),我们为每一个点对应定义一个方程。由于公式(7.1.11)是一个齐次方程,我们可以把矩阵的任意一个参数设置为1,例如:

$$e_9 = 1 \qquad (7.1.17)$$

由于这个约束,为了估计 E 矩阵最少只需要8个方程。为 $\mathbf{x}^{(j)}$ 与 $\mathbf{x}'^{(j)}$ 之间每个测量点的对应,我们可以根据公式(7.1.11)建立下面的方程:

$$\mathbf{a}_j^T \mathbf{e}' = r_j \qquad (7.1.18)$$

其中 $\mathbf{a}_j = [x'_jx_j, x'_jy_j, x'_j, y'_jx_j, y'_jy_j, y'_j, x_j, y_j, 1]^T$ 和 $\mathbf{e}' = [e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, 1]^T$。所有的点对应将导致下面的方程组：

$$[\mathbf{A}]\mathbf{e}' = \mathbf{r} \tag{7.1.19}$$

其中

$$[\mathbf{A}] = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_J]^T \tag{7.1.20}$$

然后我们可以用最小平方法使 $\|\mathbf{r}\|^2$ 最小来解公式(7.1.19)中的超定方程组。

可以证明，在解 E 矩阵时，把 E 矩阵的一个元素如公式(7.1.17)那样设置为常量，可能不会产生可达到的最小残差。一种替代的方法是要求矢量 $\mathbf{e} = [e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9]^T$ 具有单位范数，即：

$$\|\mathbf{e}\| = 1 \tag{7.1.21}$$

然后，我们可以利用一个约束的最小化问题解 E 矩阵：

$$\|[\mathbf{A}]\mathbf{e}\| \underset{\mathbf{e}}{\longrightarrow} \min \quad , \|\mathbf{e}\| = 1 \tag{7.1.22}$$

这将产生最小残差和较好的 E 矩阵的解。这个解是 $[\mathbf{A}]^T[\mathbf{A}]$ 关于最小特征值的单位特征矢量。为了避免数值不稳定性，$[\mathbf{A}]$ 的特征值和特征矢量可以用 SVD 按下式计算：

$$[\mathbf{A}] = [\mathbf{U}][\boldsymbol{\Sigma}][\mathbf{V}]^T \tag{7.1.23}$$

矩阵 $[\boldsymbol{\Sigma}]$ 只包含 $[\mathbf{A}]^T[\mathbf{A}]$ 的非负特征值的正平方根[14]。

一个测量的点对应 $(\mathbf{x}, \mathbf{x}')$ 的误差影响依赖于实际的图像坐标 $\mathbf{x}$（见习题 7.6）。靠近图像中心的对应误差并不如图像边界处的误差对解的影响大。

旋转和平移的估计 由于 E 矩阵定义为描述刚性物体的三维运动，考虑到我们只能恢复平移矢量的方位，它具有 5 个自由的角度。然而在 E 矩阵的估计中，我们允许 8 个自由度。因此从 E 矩阵提取旋转和平移，需要解一个最优化问题。

按照参考文献[21]，首先估计平移矢量。对于如公式(7.1.13)定义的 $[\mathbf{T}]$，我们有：

$$[\mathbf{T}]^T = -[\mathbf{T}] \quad 和 \quad [\mathbf{T}]\mathbf{T} = \mathbf{0} \tag{7.1.24}$$

用 $\mathbf{T}$ 乘公式(7.1.12)，我们得到：

$$[\mathbf{E}]^T \cdot \mathbf{T} = [\mathbf{R}]^T[\mathbf{T}]^T \cdot \mathbf{T} = -[\mathbf{R}]^T[\mathbf{T}] \cdot \mathbf{T} = 0 \tag{7.1.25}$$

因此，$[\mathbf{E}]$ 的所有行矢量是线性相关的 $(\det([\mathbf{E}]) = 0)$，$\mathbf{T}$ 正交于由 $[\mathbf{E}]$ 的列矢量定义的平面。如果 $\mathbf{E}$ 矩阵是根据公式(7.1.22)对有噪的点对应估计的，我们就不能假设公式(7.1.25)精确地成立。因此我们更愿意使用 SVD 根据：

$$\|[\mathbf{E}]^T\mathbf{T}\| \underset{\mathbf{T}}{\longrightarrow} \min \quad , \|\mathbf{T}\| = 1 \tag{7.1.26}$$

来估计平移矢量。类似于公式(7.1.22)中 $\mathbf{e}$ 的解，$\mathbf{T}$ 是 $[\mathbf{E}][\mathbf{E}]^T$ 的最小特征值的单位特征矢量。由公式(7.1.25)，很明显 $\mathbf{T}$ 的符号是不能确定的。类似地，$[\mathbf{E}]$ 的解不是惟一的。给定估计的 E 矩阵 $[\hat{\mathbf{E}}]$，我们知道 $-[\hat{\mathbf{E}}]$ 也是一个有效的解。在参考文献[22]中证明了可以通过选择 $\mathbf{T}$ 使下面的条件对所有的 $\mathbf{x}$ 成立来确定 $\mathbf{T}$ 的正确符号：

$$\sum_{\mathbf{x}}[\mathbf{T}[x',y',1]]^T\cdot[\mathbf{E}]\cdot[x,y,1]^T>0 \tag{7.1.27}$$

这个公式确保 X 和 X′在运动前后都位于摄像机的前面。

第二步,我们估计旋转矩阵[**R**]。公式(7.1.12)也可以写成:

$$[\mathbf{E}]^T=[\mathbf{R}]^T[\mathbf{T}]^T=-[\mathbf{R}]^T[\mathbf{T}] \tag{7.1.28}$$

因此[**R**]可以通过下面的最优化问题求出:

$$\|[\mathbf{R}]^T[-\mathbf{T}]-\mathbf{E}^T\|\xrightarrow[{[\mathbf{R}]}]{}\min\quad,[\mathbf{R}]^T=[\mathbf{R}]^{-1}\quad 和\ \det([\mathbf{R}])=1 \tag{7.1.29}$$

对于任何旋转矩阵[**R**]这个约束都是成立的。公式(7.1.29)是如下一般问题的特例:

$$\|[\mathbf{F}][\mathbf{C}]-[\mathbf{D}]\|\xrightarrow[{[\mathbf{F}]}]{}\min \tag{7.1.30}$$

其中[**F**]必须满足旋转矩阵的性质,[**C**]和[**D**]是任意的 3×3 矩阵。使用 4×4 矩阵[**B**]可以计算求解[21]:

$$[\mathbf{B}]=\sum_{i=1}^{3}[\mathbf{B}_i]^T[\mathbf{B}_i] \tag{7.1.31}$$

其中

$$[\mathbf{B}_i]=\begin{bmatrix}0 & c_{1i}-d_{1i} & c_{2i}-d_{2i} & c_{3i}-d_{3i}\\ d_{1i}-c_{1i} & 0 & -(c_{3i}+d_{3i}) & c_{2i}+d_{2i}\\ d_{2i}-c_{3i} & c_{3i}+d_{3i} & 0 & -(c_{1i}+d_{1i})\\ d_{3i}-c_{3i} & -(c_{2i}+d_{2i}) & c_{1i}+d_{1i} & 0\end{bmatrix} \tag{7.1.32}$$

这里 c_{ji}, d_{ji} 是[**C**]和[**D**]的元素。首先,我们用 SVD 确定[**B**]的最小特征值的单位特征矢量 $\mathbf{g}=[g_1,g_2,g_3,g_4]^T$。然后矩阵[**F**]可以由下式得到:

$$[\mathbf{F}]=\begin{bmatrix}g_1^2+g_2^2+g_3^2+g_4^2 & 2(g_2g_3-g_1g_4) & 2(g_2g_4+g_1g_3)\\ 2(g_2g_3+g_1g_4) & g_1^2-g_2^2+g_3^2-g_4^2 & 2(g_3g_4-g_1g_2)\\ 2(g_2g_4-g_1g_3) & 2(g_3g_4+g_1g_2) & g_1^2-g_2^2-g_3^2+g_4^2\end{bmatrix} \tag{7.1.33}$$

设[**C**] = −[**T**],[**D**] = [**E**]^T,则旋转矩阵通过[**R**] = [**F**]^T 与前面的解联系起来。

估计的精度　在本节中,我们研究估计的参数表示特征点对应的优劣程度。所得到的运动参数精度的一个合适的度量,是使用估计的旋转和平移参数根据公式(7.1.15)计算每一个特征点 **x** 的外极线。然后我们度量 **x**′与它的外极线之间的距离 d^2:

$$d^2=\frac{(\mathbf{x}'[\mathbf{E}]\mathbf{x})^2}{a(\mathbf{x},[\mathbf{E}])^2+b(\mathbf{x},[\mathbf{E}])^2} \tag{7.1.34}$$

平均距离 $\bar{d}$ 指示 E 矩阵与给定对应的近似程度。有两种可能的方法计算 $\bar{d}$。我们可以用根据公式(7.1.22)估计的 E 矩阵,或者也可以用 E 矩阵[$\hat{\mathbf{E}}$] = [**T**][**R**],它是由公式(7.1.26)和(7.1.29)估计的平移和旋转参数计算得到的,而公式(7.1.26)和(7.1.29)又是由估计的 E 矩阵计算得到的。要注意,用第二种方法计算的 $\bar{d}$ 比用第一种方法得到的 $\bar{d}$ 通常要高一个数量

级。这表明采用E矩阵的运动估计对于对应中的噪声非常敏感。先进的算法用公式(7.1.22)作为估计E矩阵的迭代算法的起始点，服从的约束是，[**E**]是由[**R**]和[**T**]通过公式(7.1.12)和(7.1.13)定义的。

物体形状的估计 知道了对应点 **x** 和 **x**′以及运动[**R**]和 **T**，我们可以根据公式(7.1.10)用两种方法计算一个点的 Z：

$$Z = \frac{T_y - y' T_z}{(x r_7 + y r_7 + r_9) y' - (x r_4 + y r_5 + r_6)} \tag{7.1.35}$$

或

$$Z = \frac{T_x - x' T_z}{(x r_7 + y r_8 + r_9) x' - (x r_1 + y r_2 + r_3)} \tag{7.1.36}$$

如果 **x**′不在外极线上，这两个方程就会得到两个不同的 Z 的估计。图 7.2 示出了公式(7.1.35)和(7.1.36)是如何把点 **x**′投影到外极线上的。确定 Z 的一个较好的方法是在 **x**′到外极线的正交投影上选择 Z。已知了 Z，用逆透视投影，由 **x** 可以确定 X 和 Y。

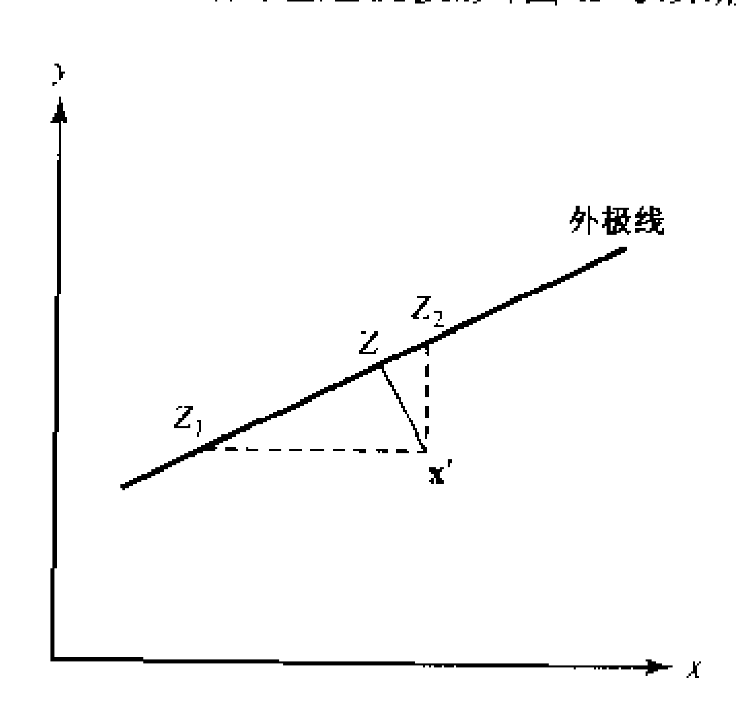


图 7.2 点 **x**′到外极线上 z 处的投影确定了三维空间中对应的物体点的位置。$\mathbf{Z}_1$ 和 $\mathbf{Z}_2$ 是用公式(7.1.35)和(7.1.36)计算的

7.2 直接运动估计

在上一节中，我们假设了对于两幅图像中可见特征的少量精确的特征点对应。由于这样的精确特征点对应不总是可以得到的，现在介绍的运动估计算法是直接使用图像信号和它的梯度。由于图像信号倾向于有噪的，我们希望用相当大量的图像点来估计三维运动，并称这些点为观测点。我们不知道观测点在前一个和后一个图像中的位置。如果我们做出的关于物体的形状、亮度和摄像机模型的假设成立，这些算法就可以度量大的运动。这些假设的一个明显的结果是一个物体在相继两幅图像中的投影是重叠的。因此直接运动估计算法的估计范围要比间接运动估计算法小。

我们将从导出图像和运动的基本关系开始(7.2.1 节)。然后，给出一个类似于7.1.2节的算法，它允许我们估计没有特征点的已知形状的三维物体的三维运动(7.2.2 节)。进一步，我

们将应用这些知识推广7.1.3节的算法，使之在没有特征点的情况下工作。在7.2.4节，我们给出几种就估计精度而言可以提高这些估计算法的健壮性的方法。

7.2.1 图像信号模型和运动

为了导出直接运动估计算法，假设两幅相继的亮度图像$\psi_k(\mathbf{x})$与$\psi_{k+1}(\mathbf{x})$之间的差只是由于物体的运动造成的。这暗示物体有漫反射的表面，并且场景是由环境光源照明的（见5.2节）。在参考文献[16]和[4]中，也考虑了照明的影响。我们现在建立两个视频帧$\psi_k(\mathbf{x})$与$\psi_{k+1}(\mathbf{x})$之间的帧差与运动的关系。

首先，我们需要图像信号的一个分析的表示。应用$\psi(\mathbf{x}+\Delta\mathbf{x})$在$\mathbf{x}$的一阶泰勒展开，可以得到：

$$\psi(\mathbf{x}+\Delta\mathbf{x}) \approx \psi(\mathbf{x}) + \frac{\partial\psi}{\partial x}\Delta x + \frac{\partial\psi}{\partial y}\Delta y \tag{7.2.1}$$

显然，这个线性信号模型只是真实图像的粗略估计。此外仅当$\Delta\mathbf{x}$很小时，这种近似才是正确的。在附录A，我们描述了几种在图像位置$\mathbf{x}$计算图像梯度$\mathbf{g}(\mathbf{x}) = [\partial\psi(\mathbf{x})/\partial x, \partial(\mathbf{x})\psi/\partial y]^T$的方法。

让我们考虑物体表面的一个观察点，它在时刻k位于$\mathbf{X}$，在时刻$k+1$位于$\mathbf{X}'$。$\mathbf{X}$投影到图像ψ_k中的$\mathbf{x}$，运动以后，同一个点$\mathbf{X}'$投影到图像ψ_{k+1}中的$\mathbf{x}'$，图像k与图像$k+1$在点$\mathbf{x}$的亮度差是：

$$\begin{aligned}\Delta\psi_k(\mathbf{x}) &= \psi_{k+1}(\mathbf{x}) - \psi_k(\mathbf{x}) \\ &= \psi_{k+1}(\mathbf{x}) - \psi_{k+1}(\mathbf{x}')\end{aligned} \tag{7.2.2}$$

公式(7.2.2)是成立的，因为根据恒定亮度假设$\psi_{k+1}(\mathbf{x}') = \psi_k(\mathbf{x})$（公式(5.2.11)），我们假设所有的变化都是由物体的运动引起的。利用公式(7.2.1)的线性信号模型和$\Delta x = \mathbf{x}' - \mathbf{x}$，可以把公式(7.2.2)重新写为：

$$\Delta\psi_k(\mathbf{x}) = -\mathbf{g}_{k+1}(\mathbf{x})\cdot(\mathbf{x}'-\mathbf{x}) \tag{7.2.3}$$

其中“.”表示矢量的内积。由于线性信号模型，$\mathbf{g}_{k+1}(\mathbf{x}) = \mathbf{g}_k(\mathbf{x})$成立。因此，公式(7.2.3)也可以表示成：

$$\Delta\psi_k(\mathbf{x}) = -\mathbf{g}_k(\mathbf{x})\cdot(\mathbf{x}'-\mathbf{x}) \tag{7.2.4}$$

注意这个公式等价于公式(6.2.3)给出的光流方程的离散形式。

公式(7.2.4)给出了图像k中的观察点$\mathbf{x}$与它在图像$k+1$中的移位位置$\mathbf{x}'$之间的关系。图7.3中示出了这种在图像梯度、帧差和二维运动之间的关系。对于图7.3(a)，公式(7.2.4)是精确成立的，因为图像信号ψ_k和ψ_{k+1}是线性的。在这个例子中，亮度差$\Delta\psi_k(\mathbf{x})$是负的，图像梯度$\mathbf{g}_k(\mathbf{x})$是正的，产生了一个正的位移$(\mathbf{x}'-\mathbf{x})>0$。图7.3(b)的情况类似；然而图像信号不再是线性的了。在这个例子中，度量的帧差$\Delta\psi_k(\mathbf{x})$小于线性信号模型所给出的帧差。因此，我们将估计出一个具有正确的符号但是幅度过小的位移。由于这个效应，我们将不得不迭代地估计运动。

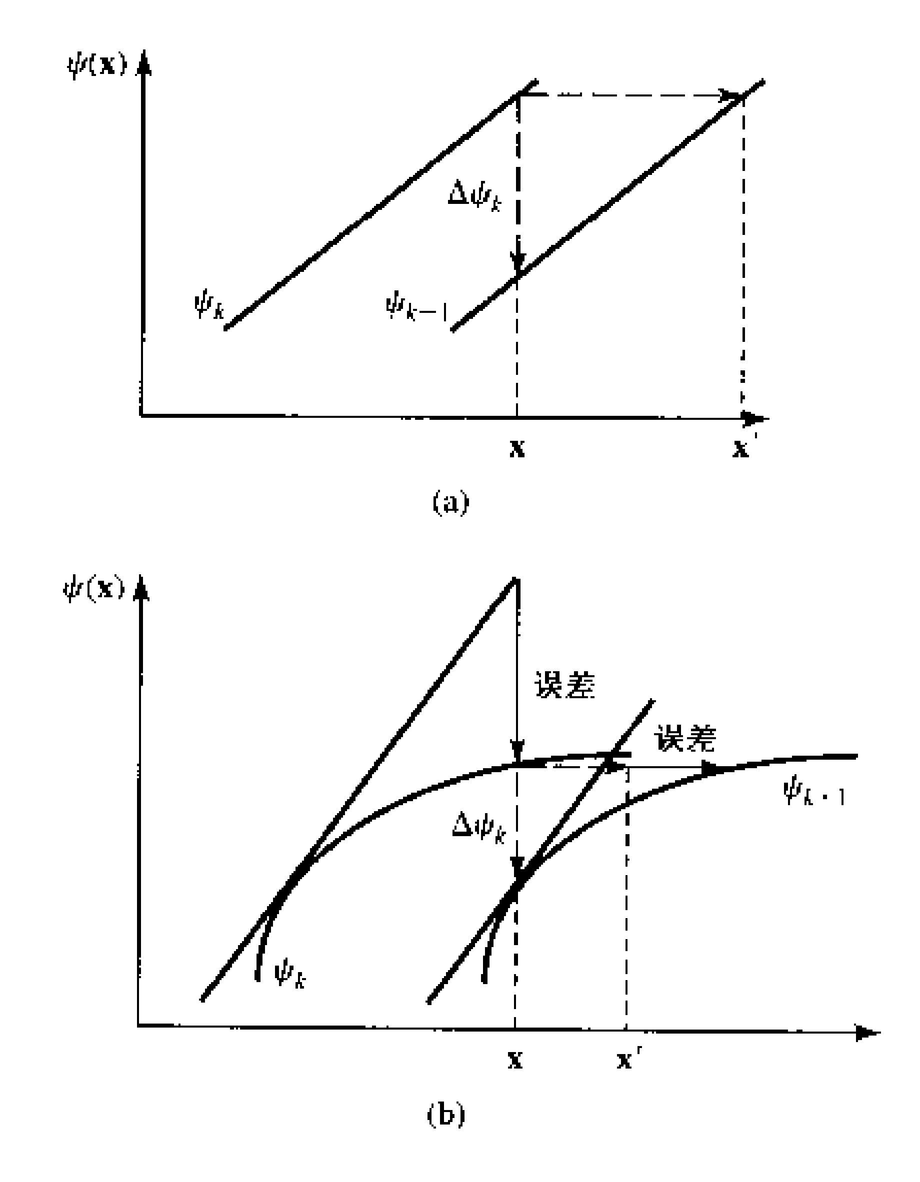


图 7.3 说明公式(7.2.4)是如何把图像信号与一个观测点的二维运动联系起来的两个例子:(a)因为线性信号模型成立,所以没有近似误差;(b)由于线性信号模型不能很好地近似图像信号,我们度量的亮度差小于所要求的,结果是估计的位移太小

观察图 7.3(b),可以想像如果图像信号包含有很高的频率,我们将会测量出一个具有不可预测的符号的亮度差。在这种情况下,位移($\mathbf{x}'-\mathbf{x}$)可能大于 $\mathbf{x}$ 附近图像信号的周期。这会造成估计的位移具有不正确的符号。因此,我们建议第一次连接迭代时,使用图像低通滤波后的版本。

下面两节描述的运动估计算法是基于图像信号模型的,以便导出帧差与三维运动的关系。这些算法用摄像机模型、运动方程和物体的形状把点 $\mathbf{x}$ 的运动与三维空间的运动联系起来。由于公式(7.2.4)是基于图像信号的线性近似的,所以估计算法将用于一个迭代估计过程之内,直到正确的运动被估计出来。为了加速算法的收敛,期望图像信号的一个高阶的近似。根据参考文献[3],我们可以用四边形图像信号模型近似图像信号,使用不明确计算二阶导数的二阶泰勒级数展开。使用这个模型,帧差近似为:

$$\Delta\psi_k(\mathbf{x}) = -\bar{\mathbf{g}}(\mathbf{x})\cdot(\mathbf{x}'-\mathbf{x}) \tag{7.2.5}$$

其中

$$\bar{\mathbf{g}}(\mathbf{x}) = \frac{1}{2}(\mathbf{g}_k(\mathbf{x}) + \mathbf{g}_{k+1}(\mathbf{x})) \tag{7.2.6}$$

平均梯度 $\bar{\mathbf{g}}(\mathbf{x})$ 考虑到了 $\psi_{k+1}(\mathbf{x})$ 的一阶和二阶梯度。由于这个四边形图像信号模型能较好地近似图像信号,产生较好的运动估计,并且易于计算,所以只要合适,我们建议使用这个模型。

7.2.2 已知形状的物体

这里推广 7.1.2 节的算法，使它在没有特征点时适用。在运动估计中，我们使用大量的观测点表示物体。由于假设我们知道物体的形状，那么这些点的三维位置是可知的。用世界坐标 $\mathbf{X}$ 替换公式(7.2.5)中的图像坐标 $\mathbf{x}$，根据公式(5.1.2)使用针孔摄像机模型，我们得到：

$$\Delta\psi_k(\mathbf{x}) = F\cdot\bar{g}_x\left(\frac{X'}{Z'}-\frac{X}{Z}\right)+F\cdot\bar{g}_y\left(\frac{Y'}{Z'}-\frac{Y}{Z}\right) \tag{7.2.7}$$

其中 $\bar{g}_x$ 和 $\bar{g}_y$ 是 $\bar{g}(\mathbf{x})$ 的 x 和 y 分量。假设物体绕着它的中心 $\mathbf{C}=[\mathbf{C}_x,\mathbf{C}_y,\mathbf{C}_x]^{\mathrm{T}}$ 旋转，并且有小的旋转角度，由公式(5.3.10)和(5.3.11)得到：

$$\mathbf{X}'=[\mathbf{R}']\cdot(\mathbf{X}-\mathbf{C})+\mathbf{C}+\mathbf{T} \tag{7.2.8}$$

把公式(7.2.8)代入公式(7.2.7)中的 $\mathbf{X}'$ 得到：

$$\begin{aligned}
\Delta\psi_k(\mathbf{x}) = {} & F\cdot\bar{g}_x/Z\cdot T_x + F\cdot\bar{g}_y/Z\cdot T_y \\
& -[(X\bar{g}_x+Y\bar{g}_y)F/Z^2+\Delta\psi_k(\mathbf{x})/Z]\cdot T_z \\
& -[[X\bar{g}_x(Y-C_y)+Y\bar{g}_y(Y-C_y)+Z\bar{g}_y(Z-C_z)]F/Z^2 \\
& +\Delta\psi(\mathbf{x})/Z(Y-C_y)]\cdot\theta_x+[[Y\bar{g}_y(X-C_x)+X\bar{g}_1(X-C_x) \\
& +Z\bar{g}_x(Z-C_z)]F/Z^2+\Delta\psi(\mathbf{x})/Z(X-C_x)]\cdot\theta_y \\
& -[\bar{g}_x(Y-C_y)-\bar{g}_y(X-C_x)]F/Z\cdot\theta_z
\end{aligned} \tag{7.2.9}$$

其中 $T_x,T_y,T_z,\theta_x,\theta_y$ 和 θ_z 是 6 个未知的运动参数，$\mathbf{X}=[X,Y,Z]^T$ 是物体表面的点，根据公式(7.2.6)，$\mathbf{g}=[\bar{g}_x,g_y]^T$ 是图像的梯度，F 是摄像机的焦距，$\mathbf{x}$ 是运动前帧 k 中的观测点的位置。

为得到 6 个运动参数的可靠估计，必须为运动物体表面上的许多观测点 $\mathbf{x}_j,j=1,2,\cdots,J$ 建立公式(7.2.9)，产生一个超定的线性方程组：

$$[\mathbf{A}]\cdot\phi-\mathbf{b}=\mathbf{r} \tag{7.2.10}$$

其中，残差 $\mathbf{r}=[r_1,\cdots,r_J]^T$，$\phi=[T_x,T_y,T_z,\theta_x,\theta_y,\theta_z]^T$，$\mathbf{b}=[\Delta\psi_k(\mathbf{x}_1),\cdots,\Delta\psi_k(\mathbf{x}_J)]^T$，$[\mathbf{A}]=[\mathbf{a}_1,\cdots,\mathbf{a}_J]^T$，而 $\mathbf{a}_j^T$ 是根据公式(7.2.9)对第 j 个观测点定义的。这个线性方程组可以通过下面的最优化问题求解：

$$\|[\mathbf{A}]\phi-\mathbf{b}\|^2=\|\mathbf{r}\|^2\xrightarrow{\phi}\min \tag{7.2.11}$$

它具有最小化在物体的观测点 $x_1,\cdots,x_J$ 上测量到的运动补偿帧差的作用。运动参数由下式给出：

$$\hat{\phi}=([\mathbf{A}]^T[\mathbf{A}])^{-1}[\mathbf{A}]^T\mathbf{b} \tag{7.2.12}$$

为了避免大矩阵求逆，我们不计算$[\mathbf{A}]$，而是直接计算 6×6 矩阵$[\mathbf{A}]^T[\mathbf{A}]$和矢量$[\mathbf{A}]^T\mathbf{b}$。

由于公式(7.2.8)和(7.2.5)的线性化，运动参数必须迭代地估计。每一次迭代以后，模型物体及其观测点都要按照非线性三维运动方程(公式(5.3.11))使用估计的运动参数 $\hat{\phi}$ 进行移动。然后，建立一套新的运动方程，给出新的运动参数更新。假设运动估计算法收敛，在迭代

的过程中,运动参数的更新会下降。由于在迭代的过程中,运动参数的更新趋向于零,由旋转矩阵线性化造成的误差也趋向于零。当残差 $\|[\mathbf{A}]\phi - \mathbf{b}\|^2$ 的下降变成可以忽略时,则迭代过程终止。

这个算法估计从帧 k 到帧 $k+1$ 物体的运动。由于这是前向运动估计,我们可以随着视频场景中物体的前向运动跟踪它。

7.2.3 平面物体

平面物体经常用做真实物体的近似(5.5.3 节)。由于这些近似通常不是用于跟踪而是用于视频编码中的运动补偿,我们现在把 7.1.3 节给出的算法推广到一个估计后向运动的直接估计方法[5]。从公式(7.1.9)开始,用下式定义图像 $k+1$ 中的点 $\mathbf{x}$ 到图像 k 中的点 $\mathbf{x}'$ 的映射 $\mathbf{A}(x,y)$:

$$\begin{aligned}\mathbf{x}' &= [x', y']^T \\ &= [A_x(\mathbf{x}), A_y(\mathbf{x})]^T \\ &= \left[\frac{a_1 x + a_2 y + a_3}{a_7 x + a_8 y + 1}, \frac{a_4 x + a_5 y + a_6}{a_7 x + a_8 y + 1}\right]^T \\ &= \mathbf{A}(\mathbf{x})\end{aligned} \tag{7.2.13}$$

并再次假设图像序列的所有变化都是由于物体的运动引起的,我们得到:

$$\psi_{k+1}(\mathbf{x}) = \psi_k(\mathbf{x}') = \psi_k(\mathbf{A}(\mathbf{x})) \tag{7.2.14}$$

类似于公式(7.2.2),帧差变成:

$$\begin{aligned}\Delta\psi_k(\mathbf{x}) &= \psi_{k+1}(\mathbf{x}) - \psi_k(\mathbf{x}) \\ &= \psi_k(\mathbf{A}(\mathbf{x})) - \psi_k(\mathbf{x})\end{aligned} \tag{7.2.15}$$

利用泰勒级数展开,我们表示在观测点 $\mathbf{x}$ 关于运动参数 a_i 的亮度函数:

$$\begin{aligned}\psi_k(\mathbf{A}(\mathbf{x})) &= \psi_k(\mathbf{x}_k) + \sum_{i=1}^{8} \frac{\partial \psi_k(\mathbf{x})}{\partial a_i}\Delta a_i \\ &\quad + \frac{1}{2}\sum_{i=1}^{8}\sum_{j=1}^{8} \frac{\partial \psi_k(\mathbf{x})}{\partial a_i}\frac{\partial \psi_k(\mathbf{x})}{\partial a_j}\Delta a_i \Delta a_j + \boldsymbol{R}_2(\mathbf{x})\end{aligned} \tag{7.2.16}$$

用 $\mathbf{a} = (a_1, \cdots, a_8)^T$, $\Delta\mathbf{a} = \mathbf{a} - \mathbf{e}$, $\mathbf{e} = (1,0,0,0,1,0,0,0)^T$, $\frac{\partial\psi_k(\mathbf{x})}{\partial a_i} = \frac{\partial\psi_k(\mathbf{x})}{\partial x}\frac{\partial A_x(\mathbf{x})}{\partial a_i} + \frac{\partial\psi_k(\mathbf{x})}{\partial y}\frac{\partial A_y(\mathbf{x})}{\partial a_i}$ 和 $\boldsymbol{R}_2(\mathbf{x})$ 表示泰勒级数展开式的高阶项。把它应用在公式(7.2.15)中得到:

$$\Delta\psi_k(\mathbf{x}) = \sum_{i=1}^{8} \frac{\partial \psi_k(\mathbf{x})}{\partial a_i}\Delta a_i + \frac{1}{2}\sum_{i=1}^{8}\sum_{j=1}^{8} \frac{\partial \psi_k(\mathbf{x})}{\partial a_i}\frac{\partial \psi_k(\mathbf{x})}{\partial a_j}\Delta a_i \Delta a_j + \boldsymbol{R}_2(\mathbf{x}) \tag{7.2.17}$$

如果四边形图像信号模型是成立的,则残差 $R_2(\mathbf{x})$ 为零,简化公式(7.2.17)为:

$$\begin{aligned}\Delta\psi_k(\mathbf{x}) &= g_x x\Delta a_1 + g_x y\Delta a_2 + g_x\Delta a_3 + g_y x\Delta a_4 + g_y y\Delta a_5 \\ &\quad + g_y\Delta a_6 - x(g_x x + \bar{g}_y y)\Delta a_7 - y(g_x x + \bar{g}_y y)\Delta a_8 \\ &= \mathbf{h}^T\Delta\mathbf{a}\end{aligned} \tag{7.2.18}$$

其中,根据公式(7.2.6),$\mathbf{g}(\mathbf{x}) = [g_x, \bar{g}_y]^T$。

对于一个物体的观测点,公式(7.2.18)定义了一个有 8 个未知运动参数的方程。为了估计运动,我们为 J 个观测点 $\mathbf{x}$ 建立方程(7.2.18),解下面的 J 阶线性方程组:

$$[\mathbf{H}]\Delta a = \Delta\psi \tag{7.2.19}$$

矩阵$[\mathbf{H}]$的行矢量 $\mathbf{h}^T$ 是根据公式(7.2.18)确定的,而根据公式(7.2.18),$\Delta\psi$ 是包含两幅图像之间帧差的矢量。用最小平方法解这个线性方程组,类似于公式(7.2.12),我们得到:

$$\Delta\mathbf{a} = ([\mathbf{H}]^T[\mathbf{H}])^{-1}[\mathbf{H}]^T\Delta\psi \tag{7.2.20}$$

由于在图像信号模型中使用的近似,这个估计过程必须迭代地实现。每次迭代后,我们把估计的运动参数用到运动补偿帧 $k+1$,使用:

$$\hat{\psi}_k(\mathbf{x}) = \psi_{k+1}(\mathbf{A}'(\mathbf{x})) \tag{7.2.21}$$

置换的帧差变成:

$$\mathrm{DFD}(\mathbf{x}) = \psi_{k+1}(\mathbf{A}'(\mathbf{x})) - \psi_k(\mathbf{x}) \tag{7.2.22}$$

其中逆映射 $\mathbf{A}'(\mathbf{x})$定义为 $\mathbf{A}'(\mathbf{A}(\mathbf{x})) = \mathbf{x}$。在迭代期间,公式(7.2.22)代替公式(7.2.15)。而且,公式(7.2.18)中的图像梯度必须用运动补偿的图像和图像 k 来计算。如果我们对全局运动参数感兴趣,那么每次迭代中估计参数就必须适当地连结起来(见习题 7.9)。

7.2.4 健壮估计

为了使运动估计不容易受图像噪声影响或避免我们的模型假设的扰乱,必须注意选择和使用观测点。有两种类型的误差会影响运动估计:由采样图像信号时出现的图像噪声引起的测量误差;当估计算法的模型假设不成立时出现的模型误差。在本节中,我们首先讨论测量误差,然后讨论模型误差以及如何使它们对估计的运动参数的影响最小化。

测量误差　前面两节的算法都是基于四边形图像信号模型的。由于这最多只在观测点位置周围的一个小区域成立,在开发一个健壮的直接运动估计器时,适当选择观测点是一个很重要的因素。这里我们给出对任何直接运动估计算法都适用的几个概念。

考虑到图像噪声,我们可以导出一个选择观测点的基本准则。假设摄像机噪声在两个相继的帧是 i.i.d,并且噪声方差为 σ_n^2,则图像差信号的噪声将是:

$$\sigma_{\Delta\psi_k}^2 = 2\cdot\sigma_n^2 \tag{7.2.23}$$

由于使用帧差计算运动,必须注意我们仅考虑了摄像机噪声影响很小的那些地方的观测点,也就是说,我们更喜欢帧差超过某个最小值的那些观测点。

根据公式(7.2.4),我们把局部位移 $\mathbf{x}'-\mathbf{x}$ 表示为一个无噪声亮度信号的函数。在有噪声的环境中,梯度 $\mathbf{g}(\mathbf{x})$应该具有大的绝对值,以便限制摄像机噪声的影响。因此我们只选择梯度大于阈值 T_G 的点作为观测点:

$$\begin{aligned}|g_x(\mathbf{x})| &> T_G\\ |g_y(\mathbf{x})| &> T_G\end{aligned} \tag{7.2.24}$$

相对大的梯度也考虑到了运动参数的精确估计,因为一个微小的误匹配立即会产生高的 DFD 或者 $\Delta\psi(\mathbf{x})$。如果我们选择的观测点具有零梯度或很小的梯度,这个点可能会移动几个像素而不会引起显著的 DFD。图 7.4 示出了一个测试序列"克莱尔"的图像和它的观测点。我们选择所有满足$|g_x(\mathbf{x})|>32$ 和$|g_y(\mathbf{x})|>32$ 的点 $\mathbf{x}$ 作为观测点。用这种观测点的初始选择,我们现在研究改进运动估计算法性能的方法。

模型误差　公式(7.2.10)和(7.2.19)是通过使残差误差的方差为最小求解的。然而这种方法对模型误差敏感[10]。模型误差的出现是因为公式(7.2.10)和(7.2.19)是基于几个模型

假设和近似的，而这些假设和近似倾向于对大多数而不是对所有观测点成立。破坏这些假设的观测点称为异点(outlier)[15]。当用最小平方法解公式(7.2.10)和(7.2.19)时，异点对解有很大的影响。因此在估计的过程中，我们必须采取措施限制这些异点的影响。下面的假设有时是不成立的：

1. 刚性的真实物体，
2. 四边形图像信号模型，
3. 模型物体形状与真实物体形状具有小的偏差，
4. 漫射照明和漫反射表面。


(a)


(b)

图 7.4 (a)测试序列“克莱尔”的图像；(b)这个图像的观测点

下面，我们讨论每一个这样的假设对运动估计精度的影响。

假设 1：在部分真实物体不是刚性的情况下(即物体是柔性的)，图像的一些区域就不能用物体的运动、形状和彩色参数来描述。这些图像区域称为模型失效区，可以由它们潜在的高帧差 $\Delta\psi(\mathbf{x})$ 加以检测。具有高帧差的观测点可以归类为异点。在迭代中，我们将只考虑那些对于：

$$\Delta\psi(\mathbf{x}) < \sigma_{\Delta\psi} \cdot T_{ST} \tag{7.2.25}$$

其中

$$\sigma_{\Delta\psi}^2 = \frac{1}{J}\sum_{j=0}^{J}(\Delta\psi(\mathbf{x}^{(j)}))^2 \tag{7.2.26}$$

成立的观测点 $\mathbf{x}$。阈值 T_{ST} 用来从考虑的问题中去掉异点。方差 $\sigma_{\Delta\psi}^2$ 应该在每次迭代后重新计算。

假设 2：根据公式(7.2.6)，运动估计是基于一个四边形信号模型的。这至多在观测点 $\mathbf{x}$ 周围的一个小区域内成立。它只允许在一次迭代中估计小的位移 $(\mathbf{x}'-\mathbf{x})$[12]。给定一个图像的梯度 $\mathbf{g}(\mathbf{x})$ 和我们希望允许估计的最大位移 d_{max}，对每个观测点可以限制所能允许的帧差 $\Delta\psi(\mathbf{x})$，并且如果需要的话，对一次给定的迭代排除一个观测点。我们假设这个点是不符合图像信号模型的。

假设 3：模型物体与真实物体之间的形状差，可以用沿着观测线的观测点的空间不确定性来描述。可以考虑在估计中使用卡尔曼(Kalman)滤波器[9]。

假设 4：尽管非漫射照明和反射的影响扰乱图像的分析，但它们通常不是单独建模的。在

一定程度上,在检测模型失效区时,这些图像区域自动被检测(见假设 1)。

综上所述,我们得到结论,应该根据公式(7.2.24)用大的图像梯度选择观测点。在运动估计中,只有通过了这个初始测试的观测点才将被考虑。对于一次给定的迭代将要使用的观测点,公式(7.2.25)提供了一种选择的准则。在参考文献[13]中,讨论了关于健壮直接运动估计和最大似然估计的进一步细节。下面,我们将简要地集中于参考文献[13]和[7]中描述的高效估计器。

除了根据公式(7.2.25)对观测点使用二进制选择准则以外,我们也可以使用连续的成本函数来控制观测点对参数估计的影响。根据公式(7.2.10),使用残差 r_j 作为观测点影响的度量[14,23]。选择适当度量的关键问题是处理异点。理想情况下,我们想要知道异点的概率密度函数。而且我们还想要异点对估计的运动参数没有影响。然而异点的概率密度函数通常是未知的。因此,如 Tukey 建议的使用成本函数 $w=(1-(r^2/b^2))^2$ 的启发式的解被求出为[7,23]:

$$r'_j = \begin{cases} r_j \cdot \left(1 - \dfrac{r_j^2}{b^2}\right) & 如果\ |r_j| < b \\ 0 & 其他 \end{cases} \tag{7.2.27}$$

代替使 $\sum_j |r_j|^2 \rightarrow \min$,我们现在令:

$$\sum_j |r_j|^2 |w_j|^2 \rightarrow \min \tag{7.2.28}$$

当 $|r|$ 降低时,成本函数 $w=(1-(r^2/b^2))^2$ 增加到 1。满足 $|r_j| \geqslant b$ 的观测点被从下一次迭代中排除。常量 b 是检测异点的阈值。为了使异点的检测自适应于图像差信号 $\Delta\psi$,我们选择 b 正比于公式(7.2.26)中的 $\sigma_{\Delta\psi}$。在运动估计中,我们把每一个观测点 $\mathbf{x}^{(j)}$ 的方程乘以它们各自的成本 $w_j=(1-(r_j^2/b^2))^2$,因此按照公式(7.2.11)和(7.2.20)自适应了它们对估计运动参数的影响。

7.3 迭代运动估计

直接运动估计算法只能估计小的运动,这主要是由于图像信号模型假设成立时所限制的范围造成的。为了估计大的运动,必须使用迭代的算法。

让我们假设从一套观测点和前面一节描述过的方程出发估计运动参数。在第一次迭代时,给定残差 r_i,我们求解 $i=1$ 时的运动参数$[\mathbf{R}_i]$和 $\mathbf{T}_i$。我们把这些运动参数用到物体及其观测点的运动补偿。如果适当,我们用非线性运动方程进行运动补偿,即使运动参数是用线性化的旋转矩阵解出的。现在我们按照与第一次迭代同样的程序建立一套新的方程,产生 $i=2$ 的运动参数$[\mathbf{R}_i]$和 $\mathbf{T}_i$。继续这些迭代,只要新的运动参数能够减小残差,即对于适当选择的阈值 T 有$(r_j^2-r_{j+1}^2)>T$。最终估计的从图像 k 到图像 $k+1$ 的运动参数,可以通过把每一次迭代的运动参数连结起来进行计算。

为了说明运动参数连结的原理,我们对于 7.2.2 节给出的算法计算连结。假设按照公式(5.3.9)运动,估计的旋转矩阵$[\hat{\mathbf{R}}]$变成:

$$[\hat{\mathbf{R}}] = \prod_{i=1}^{I} [\mathbf{R}_{I+1-i}] \tag{7.3.1}$$

其中旋转矩阵$[\mathbf{R}_i]$是在第 i 次迭代中估计的。平移矢量 $\hat{\mathbf{T}}$ 的计算必须考虑旋转中心。假设

计算在第 i 次迭代中围绕 $\mathbf{C}_i$ 的旋转，我们得到：

$$\hat{\mathbf{T}} = \mathbf{T}_l + \mathbf{C}_l + [\mathbf{R}_l](\mathbf{T}_{l-1} + \mathbf{C}_{l-1} - \mathbf{C}_l + [\mathbf{R}_{l-1}](\mathbf{T}_{l-2} + \mathbf{C}_{l-2} - \mathbf{C}_{l-1} + \cdots)) \quad (7.3.2)$$

假设旋转是绕着坐标中心的，也就是说，物体按照下式运动：

$$\mathbf{X}' = [\hat{\mathbf{R}}]\mathbf{X} + \hat{\mathbf{T}} \quad (7.3.3)$$

与间接运动估计相比较，直接运动估计通常需要更多计算量的迭代算法。然而，直接运动估计具有在估计中考虑图像信号的优点。

7.4 小结

本章讨论了几种估计三维运动的方法。我们给出了两种估计方法，间接运动估计和直接运动估计。

- **间接运动估计算法(7.1节)**依赖于运动物体的特征点。这些特征点必须在两幅图像上可靠地标示。然后我们可以由这些对应来估计三维运动。这种算法允许估计大的运动；然而，这种估计算法对特征点对应中的误差敏感。取决于我们是希望假设正交(7.1.1节)投影还是透视(7.1.2节，7.1.4节)投影，和我们是否知道物体的形状，可以采用不同的算法。如果物体在摄像机轴的方向上没有明显的运动，并且物体主要在摄像机平面内旋转，那么采用正交摄像机模型是可取的。否则，就应该采用具有透视摄像机模型的算法，它比正交模型需要更多的计算。由特征点对应估计本质矩阵(7.1.4节)使得我们能确定物体的运动以及这些特征点的三维位置。为了获得抗噪声的健壮性，在用本质矩阵确定运动和形状参数时，我们建议使用外极线约束并且使特征点到该线的距离最小化(公式(7.1.34))。
- **直接运动估计算法(7.2节)**用泰勒级数展开式近似图像信号(7.2.1节)，并且从帧差信号导出基于像素的运动(光流)。这种基于像素的运动直接用来计算三维运动参数。如果从一帧到另一帧的物体运动很小，就可以使用这种算法。通常，我们需要对三维物体的形状做出一些假设。7.2.3节描述的算法假设了一个平面物体模型。成功地运用直接运动估计算法的关键是细心选择用于测量帧差信号的观测点。在7.2.4节中，我们描述了一些方法用于在运动估计中选择这些点。
- **迭代运动估计(7.3节)**　由于直接运动估计通常只允许小的运动估计，使用迭代方法可以扩展运动估计的范围。

估计方法的选择在很大程度上依赖于可利用的测量方法。如果我们只知道少量高精度特征点的位置，则应选择间接运动估计的方法。如果我们能够用光流从图像信号可靠地估计运动，就可以使用直接运动估计。对于某些应用，这些方法的结合也许能得到最好的结果。

7.5 习题

7.1 直接与间接运动估计器之间的区别是什么？潜在的优点和缺点是什么？

7.2 从一般的三维运动方程 $\mathbf{X}' = [\mathbf{R}]\mathbf{X} + \mathbf{T}$ 和公式(7.1.8)的平面方程开始，推导公式(7.1.9)。

7.3 证明公式(7.1.10)可以写成公式(7.1.11)。

7.4 证明公式(7.1.22)的最优化问题的解是$[\mathbf{A}]^T[\mathbf{A}]$关于最小特征值的单位特征矢量。

7.5 考虑一个图像点 **x**,它是 **X** 的投影。这个点分别移动到 **x**′和 **X**′。对于运动参数 **T** 和[**R**]求出对于这个点 **x** 的外极线。对于图像点 **x** 和它的外极线,作图表示点 **X** 与摄像机中心的距离(也就是 Z 坐标)是如何决定它在外极线上的位置的,把该点置于对应 $Z=\infty$ 和 $Z=1$ 的点。

7.6 当估计 E 矩阵的时候,点对应中的测量误差 δ 的影响依赖于这些对应的绝对坐标。说明点对应的残差是如何依赖于相应点的坐标的。在公式(7.1.11)和(7.1.19)中,用 $\mathbf{x}'+\delta$ 替换 **x**′。若使一个对应的图像坐标不影响 E 矩阵的估计,那么在公式(7.1.22)中的每一个方程的加权因子是什么?(提示:最小化每一个点到外极线的距离。)

7.7 在 **x** 附近用一阶和二阶泰勒展开,给出亮度信号$\psi(\mathbf{x})$的多项式表示。

7.8 用图像信号的二阶泰勒级数展开式,推导公式(7.2.6)和公式(7.2.5)。

7.9 为了实现平面物体的直接运动估计算法(7.2.3 节),必须知道 **A** 的逆映射。证明映射 **A** 满足一组四个公理(即闭包、存在一个逆、存在一个单位元素、结合律),计算映射 **A** 的逆以及两次迭代以后运动参数估计的和。

7.10 讨论提高直接运动估计算法的健壮性的方法。

7.6 文献目录

[1] Aizawa, K., H. Harashima, and T. Saito. Model-based analysis-synthesis image coding (MBASIC) system for a person's face. *Signal Processing: Image Commun.* (Oct. 1989), 1(2):139–52.

[2] Beardsley, P., P. H. S. Torr, and A. Zisserman. 3-D model aquisition from extended image sequences. In B. Buxton and Cipolla R., eds., *Proceedings of the 4th European Conference on Computer Vision, LNCS 1065,* 683–95. Cambridge, England: Springer-Verlag, 1996.

[3] Bierling, M. Displacement estimation by hierarchical block matching. In *SPIE: Visual Commun. Image Processing* (Nov. 1988), SPIE-1001:942–51.

[4] Bozdagi, G., A. M. Tekalp, and L. Onural. An improvement to MBASIC algorithm for 3-D motion and depth estimation. *IEEE Trans. Image Process.* (June 1994), 3:711–16.

[5] Hötter, M., and R. Thoma. Image segmentation based on object oriented mapping parameter estimation. *Signal Processing* (Oct. 1988), 15(3):315–34.

[6] Huang, T. S., and A. N. Netravali. Motion and structure from feature correspondences: A review. *Proceedings of the IEEE* (Feb. 1994), 82(2):252–68.

[7] Li, H., and R. Forchheimer. Two-view facial movement estimation. *IEEE Trans. on Circuits and Systems for Video Technology* (June 1994), 4(3):276–87.

[8] Longuet-Higgins, H. C. A computer algorithm for reconstructing a scene from two projections. *Nature* (Sept. 1981), 293(5828):133–35.

[9] Martinez, G. Shape estimation of articulated 3-D objects for object-based analysis-synthesis coding (OBASC). *Signal Processing: Image Communication* (March 1997), 9(3):175–99.

[10] Meer, P., D. Mintz, and D. Y. Kim. Robust regression methods in computer vision:

a review. *International Journal of Computer Vision* (April 1991), 6(1):59–70.

[11] Moons, T. A guided tour through multiview relations. In R. Koch and L. van Gool, eds., *Proceedings SMILE Workshop, LNCS 1506*, 304–46. Berlin: Springer-Verlag, 1998.

[12] Musmann, H. G., P. Pirsch, and H.-J. Grallert. Advances in picture coding. *Proceedings of the IEEE* (April, 1985), 73(4):523–48.

[13] Ostermann, J. Object-based analysis-synthesis coding based on moving 3-D objects. In L. Guan, S. Y. Kung, and J. Larsen, eds., *Multimedia Image and Video Processing*, 289–329. New York: CRC Press, 2000.

[14] Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992.

[15] Rousseeuw, P. J., and A. M. Leroy. *Robust Regression and Outlier Detection*. New York: John Wiley & Sons, 1987.

[16] Stauder, J. Estimation of point light source parameters for object-based coding. *Signal Processing: Image Communication* (Nov. 1995), 46(7):355–79.

[17] Tsai, R. Y., and T. S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Trans. Pattern Anal. Machine Intell.* (Jan. 1981), 6:13–16.

[18] Tsai, R. Y., and T. S. Huang. Estimating three-dimensional motion parameters of a rigid planar patch. *IEEE Transactions on Acoustics, Speech and Signal Processing* (Dec. 1981), 29(6):1147–52.

[19] Tsai, R. Y., and T. S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Jan. 1984), 6(1):13–27.

[20] Tsai, R. Y., T. S. Huang, and W.-L. Zhu. Estimating three-dimensional motion parameters of a rigid planar patch. II. Singular value decomposition. *IEEE Transactions on Acoustics, Speech and Signal Processing* (Aug. 1982), 30(4):525–34.

[21] Weng, J., T. S. Huang, and N. Ahuja. Motion and structure from two perspective views: algorithms, error analysis, and error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (May 1989), 11(5):451–76.

[22] Zhuang, X. A simplification to linear 2-view motion algorithms. *Computer Vision Graphics and Image Processing* (1989), 46(2):175–78.

[23] Zhuang, X., T. Wang, and P. Zhang. A highly robust estimator through partially likelihood function modeling and its application in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligience* (Jan. 1992), 14(1):19–34.

第 8 章　视频编码基础

视频编码是数字视频处理的重要应用。视频编码的目的是要减少视频序列的码率，以便能够在给定的通信信道上实时传输视频。信道带宽因应用和传输媒体的不同而异。对于使用常规电话线路的可视电话应用系统，可利用 20 kbps 的视频编码。对于标准清晰度卫星广播电视信号，可利用 6 Mbps 的码率。除了通信应用系统外，储存和检索也需要视频编码，不同存储介质有不同的容量和存取速率，因此需要不同的压缩量。由于这种很宽的码率范围，已经开发了不同类型的算法。第一类算法允许对任意视频信号进行有效编码而不需要分析视频内容，第二类算法识别视频序列中的区域和物体并对它们进行编码。我们称前者为基于波形的视频编码器，而后者为基于内容的视频编码器。

在本章中，我们首先在 8.1 节中综述各种基于波形的和基于内容的视频编码，然后在 8.2 节中回顾概率和信息论的基本概念；8.3 节介绍用于无损编码最优化的信息理论和用于有损编码最优化的率失真理论；接下来叙述无损编码(8.4 节)和最基本的有损编码技术，包括标量量化和矢量量化(8.5 节和 8.6 节)。第 9 章和第 10 章分别讨论基于波形和基于内容的视频编码技术。

8.1　编码系统概述

8.1.1　一般结构

视频编码算法的组成在很大程度上是由视频序列建模所采用的信源模型确定的。视频编码器寻求用它的信源模型描述视频序列的内容。信源模型可做出图像序列的像素之间在时间和空间上相关性的假设，它也可考虑物体的形状和运动或照度的影响。在图 8.1 中，我们给出了一个视频编码系统的基本组成。在编码器中，首先用信源模型的参数描述数字化的视频序列。如果我们使用像素统计独立的信源模型，那么这种信源模型的参数就是每个像素的亮度和色度的幅度。另一方面，如果我们使用把一个场景描述成几个物体的模型，那么参数就是各个物体的形状、纹理和运动。下一步，信源模型参数被量化成有限的符号集。量化参数取决于比特率与失真间所期望的折中。最后，用无损编码技术把量化参数映射成二进制码字；这种技术进一步利用量化参数的统计特性。产生的比特流在通信信道上传输。解码器反向进行编码器的二进制编码和量化过程，重新得到信源模型的量化参数。然后，解码器的图像合成算法用信源模型的量化参数计算解码的视频帧。

8.1.2　视频编码方案分类

在本小节中，我们对几种流行的视频编码算法提供一个综述，并把它们放入相应的信源模型中。一个编码算法的信源模型要根据其编码参数集和图像合成算法确定，图像合成算法是根据解码参数构成解码图像。

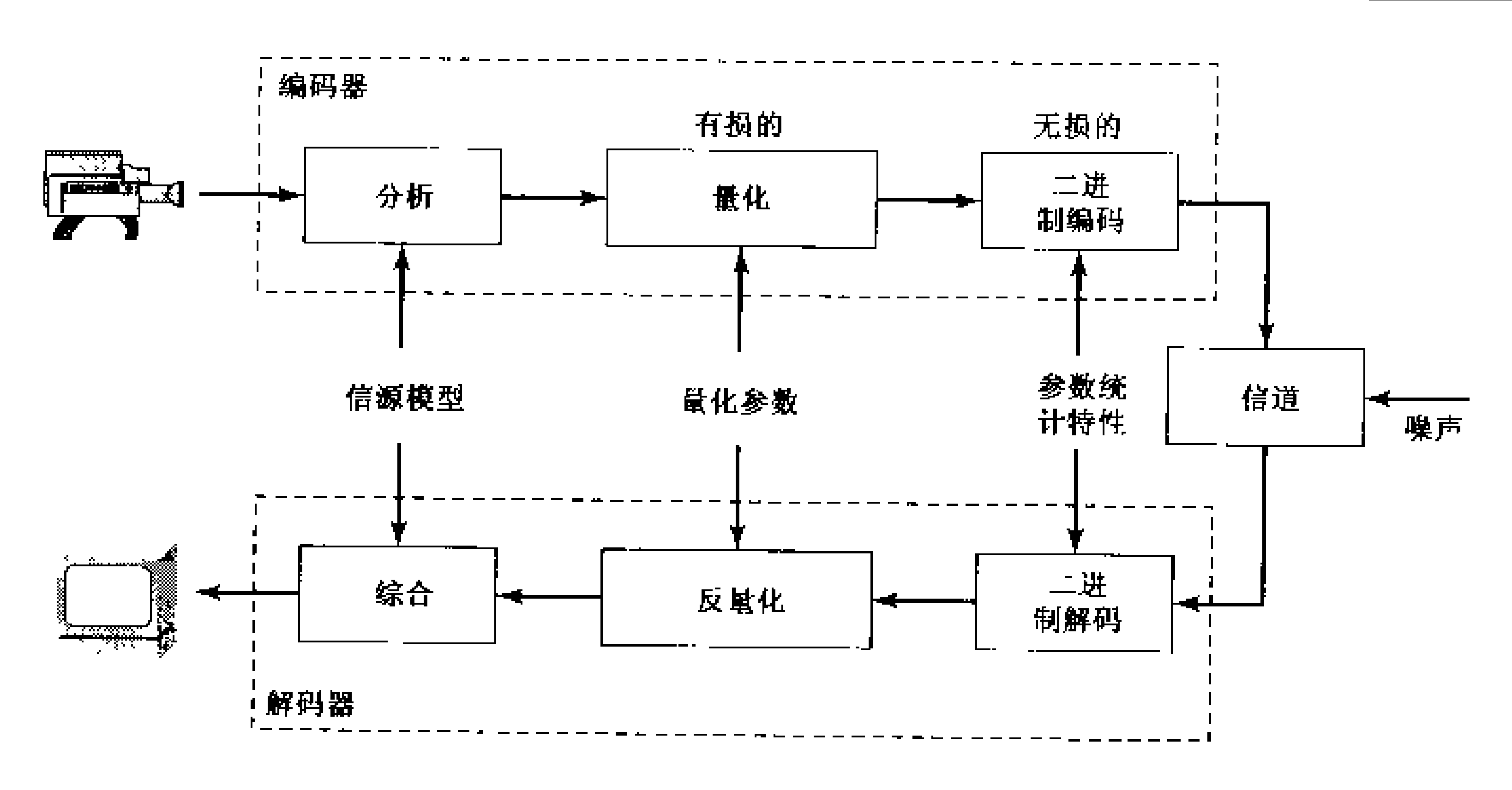


图 8.1 视频编码系统综述

基于波形的编码 第一类中的技术都试图尽可能准确地表示各个像素的颜色值，而不考虑一组像素可以表示一个物理物体这一事实。

把像素假设为统计上独立的，这样得到的模型是最简单的信源模型（表 8.1）。相关的编码技术就称为脉冲编码调制（PCM，pulse coded modulation）。图像信号的 PCM 表示通常不用于视频编码，因为与其他信源模型相比它的效率较低。

在大多数图像中，我们发现邻近像素的颜色相关性很高。为了减少比特率，最好用变换来利用这种性质，如 Karhunen-Loeve 变换（KLT）、离散余弦（DCT）或小波变换（将在第 9 章和第 11 章中介绍）。变换旨在去除原样点值间的相关性，并把原始信号的能量集中到几个系数上。需要量化和编码的参数是变换系数。利用相邻样点间相关性的另一种方法是预测编码；这种方法是先由前面编码的样点预测要编码的样点值，然后对预测误差进行量化和编码，预测误差与原始信号相比具有较小的相关性和较低的能量。变换编码和预测编码都可看做是矢量量化的一种特殊情况，矢量量化一次量化一个样点块（一个矢量）。从本质上说，它寻找出现在信号中的典型块模式，并用典型模式之一来近似任何一个块。

如今的视频编码标准 H.261，H.263，MPEG-1，MPEG-2 和 MPEG-4（将在第 13 章中讨论）都采用基于块的混合编码方法；它综合了预测编码和变换编码（第 9 章）。这种编码技术把每幅图像分成固定大小的块。第 k 帧的每个块用前面第 $k-1$ 帧的一个已位移位置处的相同尺寸的块合成得到。对第 k 帧的所有块都这样做。这样产生的图像称为预测图像。编码器把所有块的二维运动矢量传送到解码器，以便解码器能够计算同样的预测图像。编码器从原始图像中减去这幅预测图像，得到的就是预测误差图像。如果用预测图像来合成一个块不够准确——也就是说，如果块的预测误差超出某个阈值——那么编码器就用变换编码把这个块的预测误差传送到解码器。解码器把预测误差与预测图像相加，从而合成解码图像。因此，基于块的混合编码是基于平移的运动块信源模型的。除了颜色信息编码为预测误差的变换系数外，还必须传输运动矢量。值得注意的是，这种编码可切换到较简单的像素统计相关的信源模型。每当不涉及前面一帧就能更有效地完成块的编码时就进行这种切换。

表 8.1　信源模型、参数集和编码技术的比较

信源模型	编码参数	编码技术
统计独立的像素	每个像素的颜色	PCM
统计相关的像素	每个块的颜色	变换编码、预测编码和矢量量化
平移运动的块	每个块的颜色和运动矢量	基于块的混合编码
运动的未知物体	每个物体的形状、运动和颜色	分析与合成编码
运动的已知物体	每个已知物体的形状、运动和颜色	基于知识的编码
已知行为的运动已知物体	每个物体的形状、颜色和行为	语义编码

基于内容的编码　上述基于块的混合编码技术实际上是用固定大小的方块来近似场景中物体的形状。因此在物体边界上的块中它们会产生高预测误差。这些边界块包含具有不同运动的两种物体,因此用一个运动矢量就不能说明两个不同的运动。基于内容的编码器认识到这样的问题,企图把视频帧分成对应于不同物体的区域,并分别编码这些物体。对于每个物体,除了运动和纹理信息外还必须传输形状信息。

在基于物体的分析与合成编码(OBASC,将在第 10 章中讨论)中,通过模型物体描述视频场景的每个运动物体。为了描述物体的形状,分析与合成编码采用分割算法。此外,还估计每个物体的运动和纹理参数。在最简单情况下,二维轮廓描述物体形状,运动矢量场描述它的运动,而颜色波形描述它的纹理。其他方法用三维线框描述物体。用第 $k-1$ 帧中物体的形状和颜色以及形状和运动的更新参数来描述第 k 帧中的物体。解码器用当前运动和形状参数以及前一帧的颜色参数合成物体。只对那些图像合成失败的图像区域,才传输颜色信息。

在视频序列中的物体种类已知的情况下,可采用基于知识的编码,这种编码使用特别设计的线框来描述已识别出的物体类型(第 10 章)。已经开发了几种用预定义的线框来编码人头的方法。使用预定义线框可增加编码效率,因为它自适应于物体的形状。有时,也把这种技术称为基于模型的编码。

当已知可能的物体类型和它们的行为时,可以用语义编码。例如,对于一个人脸物体,"行为"指的是与特殊面部表情相关的一系列面部特征点的时间轨迹。人脸的可能行为包括典型面部表情,诸如高兴、悲伤、生气等。在这种情况下,估计描述物体行为的参数并传输给解码器。这种编码方法有达到非常高的编码效率的潜力,因为物体(如脸)可能的行为数目非常小,所以说明行为所需的比特数比用传统的运动和颜色参数描述实际动作所需的比特数少得多。

8.2　概率论和信息论中的基本概念

在视频编码或任何信号编码中,把给定信号作为随机过程的一个实现。在信源编码理论中,把随机过程作为信源。编码方法的效率取决于如何充分利用信源的统计特性。在介绍信源编码理论和技术前,首先回顾如何用概率分布表示随机过程,并从信息理论中引出熵和互信息的概念。

8.2.1　平稳信源的特征

我们只考虑定义在离散索引集上的信源,从而信源是一个随机序列。用$\mathcal{F}=\{\mathcal{F}_n\}$表示信源,其中$\mathcal{F}_n$表示对应于第 n 个样点的随机变量(RV)。在给定的实现中,$\mathcal{F}_n$所取的实际值由

f_n 表示。注意，对于多维空间上的离散信号，既可以把它排列成一维序列，这时 n 就表示样点的一维索引；也可以假设 n 是多维索引。同样，$\mathcal{F}_n$ 可以是标量也可以是矢量。在后一种情况下，$\mathcal{F}_n$ 也称为随机矢量。如果$\mathcal{F}_n$ 仅可从一个有限的字母表$\mathcal{A}=\{a_1,a_2,\cdots,a_L\}$中取符号，那么就称$\mathcal{F}_n$ 为离散随机变量，而称$\boldsymbol{\mathcal{F}}$为离散信源①。

作为例子考虑一个数字彩色视频。在此情况下，索引 n 表示像素位置和帧数的特殊组合，而$\mathcal{F}_n$ 是三维随机向量，表示第 n 个像素所取的三基色值。因为每个颜色值都量化为 256 级，所以数字视频是一个字母表大小为 256^3 的离散信源。任何给定的视频序列都是这个离散随机过程的一个特殊实现。另一方面，模拟视频是一个连续空间的连续幅度的随机过程。采样过程把它转换成一个离散空间的连续幅度的过程。只有在量化后，它才变成离散空间的离散幅度的过程(即离散信源)。在真实世界中，通常用采样和量化的方法就可以由连续空间连续幅度的信源获得离散信源。

在本节中，我们仅考虑平稳过程；在平稳过程中，$\mathcal{F}_n$ 的概率分布不取决于索引 n，而且对于索引中的共同位移，一组 N 个样点的联合分布不变。我们用 $p_{\mathcal{F}_n}(f)$表示在离散信源情况下的概率质量函数(pmf)或在连续信源情况下的概率密度函数(pdf)。另外，我们用 $p_{\mathcal{F}_{n+1},\mathcal{F}_{n+2},\cdots,\mathcal{F}_{n+N}}(f_1,f_2,\cdots,f_N)$表示$\boldsymbol{\mathcal{F}}$中任何 N 个相继样点的联合概率质量函数或概率密度函数。我们也用 $p_{\mathcal{F}_n|\mathcal{F}_{n-1},\mathcal{F}_{n-2},\cdots,\mathcal{F}_{n-M}}(f_{M+1}|f_M,f_{M-1},\cdots,f_1)$表示在给定前面 M 个样点时任何样点$\mathcal{F}_n$ 的条件概率质量函数或概率密度函数。当隐含的上下文关系很清楚时，我们就简单地用 $p(f)$，$p(f_1,f_2,\cdots,f_N)$和 $p(f_{M+1}|f_M,f_{M-1},\cdots,f_1)$表示上述函数。

一类重要的平稳信源称为独立恒等分布(i.i.d.)信源，它满足 $p(f_1,f_2,\cdots,f_N)=p(f_1)p(f_2)\cdots p(f_N)$和 $p(f_{M+1}|f_M,f_{M-1},\cdots,f_1)=p(f_{M+1})$。i.i.d. 信源也称为无记忆的。另一类重要的平稳信源是马尔可夫过程，在这个过程中一个样点仅取决于它的前一个样点，也就是说，$p(f_{M+1}|f_M,f_{M-1},\cdots f_1)=p(f_{M+1}|f_M)$。更一般地，一个 M 阶马尔可夫过程是，在这个过程中一个样点仅取决于它前面的 M 个样点。任何 N 个样点符合 N 阶高斯分布的过程称为高斯过程。如果两个样点$\mathcal{F}_n$ 和$\mathcal{F}_m$ 之间协方差的形式是$C(\mathcal{F}_n,\mathcal{F}_m)=\sigma^2\rho^{-(n-m)}$，一个高斯过程是马尔可夫过程。在过程是二维的情况下，这样的过程称为高斯-马尔可夫过程或高斯-马尔可夫场(GMF)。在图像和视频处理中，实际图像或视频帧常以 GMF 为模型。

8.2.2 离散信源的熵和互信息

本节介绍信息理论中的两个非常重要的概念：熵和互信息。正如将要看到的，它们分别提供用无损和有损编码所能达到的最小比特率的界限。我们首先定义离散 RV 的熵和互信息，然后再把这些定义应用于离散信源的样点。

定义 8.1(离散 RV 的熵) 具有字母表$\mathcal{A}$和概率质量函数 $p_{\mathcal{F}}(f)$的离散随机变量 $\mathcal{F}$的熵定义为：

$$H(\mathcal{F})=-\sum_{f\in\mathcal{A}}p_{\mathcal{F}}(f)\log_2 p_{\mathcal{F}}(f) \tag{8.2.1}$$

① 注意：在本章中，"离散"和"连续"这两个词是指样点幅度，而在前几章中是指定义信号的空间。数字信号是一个离散空间信号，它的每个采样仅取离散值。

通常，log 可取任何底数。在此仅考虑以 2 为底的情况，以便用比特表示熵。同时假设 $0\log 0 = 0$，以便把零概率的符号加到字母表中不会改变熵。

离散 RV 的熵总是非负的，因为对任何概率质量函数都有 $0 \leqslant p(f) \leqslant 1$。在字母表长为 L 的所有 RV 中，均匀分布 $p(f) = 1/L, f \in \mathcal{A}$ 达到最大熵，$H_{\max} = \log_2 L$。

熵是关于 RV $\mathcal{F}$ 不确定性的测度。它取决于 $\mathcal{F}$ 的概率质量函数，而不取决于 $\mathcal{F}$ 所用的实际字母表。如果 $\mathcal{F}$ 能够以等概率取 $\mathcal{A}$ 中的任何值，它就最不确定，因此它具有最大熵。另一方面，如果 $\mathcal{F}$ 以概率 1 取 $\mathcal{A}$ 中的一个特殊符号，则它没有不确定性，因此熵就为零。RV 的不确定性又可以认为是能够被 RV 所携带的信息：当向一个人出示 $\mathcal{F}$ 所取的实际值时，第一种情况可以比第二种情况告诉这个人更多的信息。

定义 8.2（两个离散 RV 的联合熵）　令 $\mathcal{F}$ 和 $\mathcal{G}$ 表示两个离散 RV，具有联合概率质量函数 $p_{\mathcal{F},\mathcal{G}}(f,g), f \in \mathcal{A}_f, g \in \mathcal{A}_g$。它们的联合熵定义为：

$$H(\mathcal{F},\mathcal{G}) = -\sum_{f \in \mathcal{A}_f}\sum_{g \in \mathcal{A}_g} p_{\mathcal{F},\mathcal{G}}(f,g)\log_2 p_{\mathcal{F},\mathcal{G}}(f,g) \tag{8.2.2}$$

定义 8.3（两个离散 RV 之间的条件熵）　令 $\mathcal{F}$ 和 $\mathcal{G}$ 表示两个离散 RV，具有边际概率质量函数 $p_{\mathcal{G}}(g)$ 和条件概率质量函数 $p_{\mathcal{F}|\mathcal{G}}(f|g), f \in \mathcal{A}_f, g \in \mathcal{A}_g$。在给定 $\mathcal{G}$ 的条件下，$\mathcal{F}$ 的条件熵定义为：

$$\begin{aligned} H(\mathcal{F} \mid \mathcal{G}) &= \sum_{g \in \mathcal{A}_g} p_{\mathcal{G}}(g) H(\mathcal{F} \mid g) \\ &= -\sum_{g \in \mathcal{A}_g} p_{\mathcal{G}}(g) \sum_{f \in \mathcal{A}_f} p_{\mathcal{F}|\mathcal{G}}(f \mid g)\log_2 p_{\mathcal{F}|\mathcal{G}}(f \mid g) \end{aligned} \tag{8.2.3}$$

定义 8.4（两个离散 RV 之间的互信息）　令 $\mathcal{F}$ 和 $\mathcal{G}$ 表示两个离散 RV，具有联合概率质量函数 $p_{\mathcal{F},\mathcal{G}}(f,g), f \in \mathcal{A}_f, g \in \mathcal{A}_g$，以及边际分布 $p_{\mathcal{F}}(f)$ 和 $p_{\mathcal{G}}(g)$。$\mathcal{F}$ 与 $\mathcal{G}$ 之间的互信息定义为：

$$I(\mathcal{F};\mathcal{G}) = \sum_{f \in \mathcal{A}_f}\sum_{g \in \mathcal{A}_g} p_{\mathcal{F},\mathcal{G}}(f,g)\log_2 \frac{p_{\mathcal{F},\mathcal{G}}(f,g)}{p_{\mathcal{F}}(f)p_{\mathcal{G}}(g)} \tag{8.2.4}$$

$\mathcal{F}$ 与 $\mathcal{G}$ 之间的互信息确定 $\mathcal{G}$ 关于 $\mathcal{F}$ 所提供的信息量。换句话说，在给定 $\mathcal{G}$ 的一个实现时，它度量确定 $\mathcal{F}$ 所需的比特数的减少。互信息总是非负的。容易证明：

$$I(\mathcal{F};\mathcal{G}) = I(\mathcal{G};\mathcal{F}) \geqslant 0 \tag{8.2.5}$$

熵、条件熵和互信息之间的关系　下面总结了熵、条件熵和互信息之间的一些重要关系。很容易证明这些关系（习题 8.2）。

$$H(\mathcal{F},\mathcal{G}) = H(\mathcal{G}) + H(\mathcal{F} \mid \mathcal{G}) = H(\mathcal{F}) + H(\mathcal{G} \mid \mathcal{F}) \tag{8.2.6}$$

$$I(\mathcal{F};\mathcal{G}) = H(\mathcal{F}) - H(\mathcal{F} \mid \mathcal{G}) = H(\mathcal{G}) - H(\mathcal{G} \mid \mathcal{F}) \tag{8.2.7}$$

$$I(\mathcal{F};\mathcal{G}) = H(\mathcal{F}) + H(\mathcal{G}) - H(\mathcal{F},\mathcal{G}) \tag{8.2.8}$$

$$I(\mathcal{F};\mathcal{G}) \leqslant H(\mathcal{F}), \qquad I(\mathcal{F};\mathcal{G}) \leqslant H(\mathcal{G}) \tag{8.2.9}$$

$$H(\mathcal{F}) \geqslant H(\mathcal{F} \mid \mathcal{G}), \qquad H(\mathcal{G}) \geqslant H(\mathcal{G} \mid \mathcal{F}) \tag{8.2.10}$$

$$H(\mathcal{F},\mathcal{G}) \leqslant H(\mathcal{F}) + H(\mathcal{G}) \tag{8.2.11}$$

$$I(\mathcal{F};\mathcal{F}) = H(\mathcal{F}) = H(\mathcal{F},\mathcal{F}) \tag{8.2.12}$$

公式(8.2.10)表明条件作用减小了 RV 的熵（即不确定性）。从公式(8.2.7)可看到，$I(\mathcal{F};\mathcal{G})$ 描述由知道 $\mathcal{G}$ 而引起的 $\mathcal{F}$ 不确定性（或信息）的减小。

上述定义是对一个 RV 或两个 RV 之间给出的。对于随机矢量也可类似地定义它们。把这些定义应用到由离散平稳信源的相继样点组成的随机矢量，我们得到如下的定义。

定义 8.5(离散信源的熵) 具有 N 阶联合概率质量函数 $p(f_1,f_2,\cdots,f_N)$ 的离散平稳信源 $\mathcal{F}$ 的 N 阶熵是 $\mathcal{F}$ 的 N 个相继样点 $\mathcal{F}_1,\mathcal{F}_2,\cdots,\mathcal{F}_N$ 之间的联合熵：

$$\begin{aligned}H_N(\mathcal{F}) &= H(\mathcal{F}_1,\mathcal{F}_2,\cdots,\mathcal{F}_N)\\ &= -\sum_{[f_1,f_2,\cdots,f_N]\in\mathcal{A}^N} p(f_1,f_2,\cdots,f_N)\log_2 p(f_1,f_2,\cdots,f_N)\end{aligned}\tag{8.2.13}$$

其中，$\mathcal{A}^N$ 表示 $\mathcal{A}$ 的 N 重笛卡儿(Cartesian)积。

定义 8.6(离散信源的条件熵) 具有 M 阶联合概率质量函数 $p(f_1,f_2,\cdots,f_M)$ 和 M 阶条件概率质量函数 $p(f_{M+1}\mid f_M,f_{M-1},\cdots,f_1)$ 的离散平稳信源 $\mathcal{F}$ 的 M 阶条件熵，是在给定样点 $\mathcal{F}_{M+1}$ 前面的 M 个样点 $\mathcal{F}_M,\mathcal{F}_{M-1},\cdots,\mathcal{F}_1$ 的条件下 $\mathcal{F}_{M+1}$ 的条件熵：

$$\begin{aligned}H_{C,M}(\mathcal{F}) &= H(\mathcal{F}_{M+1}\mid\mathcal{F}_M,\mathcal{F}_{M-1},\cdots,\mathcal{F}_1)\\ &= \sum_{[f_1,f_2,\cdots,f_M]\in\mathcal{A}^M} p(f_1,f_2,\cdots,f_M)H(\mathcal{F}_{M+1}\mid f_M,f_{M-1},\cdots,f_1)\end{aligned}\tag{8.2.14}$$

其中

$$\begin{aligned}&H(\mathcal{F}_{M+1}\mid f_M,f_{M-1},\cdots,f_1)\\ &= -\sum_{f_{M+1}\in\mathcal{A}} p(f_{M+1}\mid f_M,f_{M-1},\cdots,f_1)\log_2 p(f_{M+1}\mid f_M,f_{M-1},\cdots,f_1)\end{aligned}\tag{8.2.15}$$

定义 8.7(离散信源的熵率) 可以证明 $\frac{1}{N}H_N(\mathcal{F})$ 和 $H_{C,N}(\mathcal{F})$ 都是 N 的非递增函数。而且，$\lim_{N\to\infty}\frac{1}{N}H_N(\mathcal{F})$ 和 $\lim_{N\to\infty}H_{C,N}(\mathcal{F})$ 都存在且相等。这个极限定义为该信源的熵率：

$$\bar{H}(\mathcal{F}) = \lim_{N\to\infty}\frac{1}{N}H_N(\mathcal{F}) = \lim_{N\to\infty}H_{C,N}(\mathcal{F})\tag{8.2.16}$$

联合熵、条件熵和熵率之间的关系 可以证明(习题 8.3)对于任何有限的 N 和 M 有：

$$H_N(\mathcal{F}) = \sum_{m=1}^{N-1}H_{C,m}(\mathcal{F}) + H_1(\mathcal{F})\tag{8.2.17}$$

$$\bar{H}(\mathcal{F}) \leqslant H_{C,N-1}(\mathcal{F}) \leqslant \frac{1}{N}H_N(\mathcal{F}) \leqslant H_1(\mathcal{F})\tag{8.2.18}$$

如果 $\mathcal{F}$ 是独立恒等分布，那么：

$$\bar{H}(\mathcal{F}) = \frac{1}{N}H_N(\mathcal{F}) = H_{C,M}(\mathcal{F}) = H_1(\mathcal{F})\tag{8.2.19}$$

如将在 8.3.1 节中所说明的，熵率是无损地编码离散信源所需要的最小比特率的下限。只有当无数的样点一起编码时才能满足此条件。另一方面，当一个样点独立地编码或以前面的样点为条件进行编码时，一阶熵和条件熵分别提供可达到的比特率下限。

定义 8.8(离散信源之间的互信息) 令 $\mathcal{F}$ 和 $\mathcal{G}$ 分别表示具有 N 阶概率质量函数 $p(f_1,f_2,\cdots,f_N)$ 和 $p(g_1,g_2,\cdots,g_N)$ 的两个离散平稳信源，并令 $p(f_1,f_2,\cdots,f_N,g_1,g_2,\cdots,g_N)$ 表示 $\mathcal{F}$ 的 N 个样点与 $\mathcal{G}$ 的 N 个样点的联合概率质量函数。$\mathcal{F}$ 与 $\mathcal{G}$ 之间的 N 阶互信息定义为：

$$\begin{aligned}I_N(\mathcal{F};\mathcal{G}) = &\sum_{[f_1,f_2,\cdots,f_N]\in\mathcal{A}_f^N}\ \sum_{[g_1,g_2,\cdots,g_N]\in\mathcal{A}_g^N} p(f_1,f_2,\cdots,f_N,g_1,g_2,\cdots,g_N)\\ &\cdot\log_2\frac{p(f_1,f_2,\cdots,f_N,g_1,g_2,\cdots,g_N)}{p(f_1,f_2,\cdots,f_N)p(g_1,g_2,\cdots,g_N)}\end{aligned}\tag{8.2.20}$$

在有损信源编码中，信源$\mathcal{F}=\{\mathcal{F}_n\}$由量化的形式$\mathcal{G}=\{\mathcal{G}_n\}$表示。一阶互信息 $I_1(\mathcal{F};\mathcal{G})$度量量化后的样点$\mathcal{G}_n$关于原始样点$\mathcal{F}_n$所提供的信息量。另一方面，$I_N(\mathcal{F};\mathcal{G})$度量具有 N 个量化样点的块关于原始 N 个样点所提供的信息。将在 8.3.2 节中说明，$\lim_{N\to\infty}\min\frac{1}{N}I_N(\mathcal{F};\mathcal{G})$提供$\mathcal{F}$和$\mathcal{G}$之间所期望的失真 D 所需的最小比特率下限，这里是在相对于$\mathcal{F}$失真为 D 的所有可能$\mathcal{G}$上取最小。

8.2.3　连续信源的熵和互信息

对于连续信源，在支撑区内任何可能值的概率质量函数为零。直接应用前面定义的熵就会得到无穷熵。换一种办法，我们用信源的概率密度函数来定义微分熵①。

定义 8.9(连续 RV 的微分熵)　具有支撑区$\mathcal{B}$和概率密度函数 $p_{\mathcal{F}}(f)$的连续随机变量 $\mathcal{F}$ 的微分熵定义为：

$$h(\mathcal{F})=-\int_{f\in\mathcal{B}}p_{\mathcal{F}}(f)\log_2 p_{\mathcal{F}}(f)\,df \tag{8.2.21}$$

与熵一样，微分熵度量连续 RV 的不确定性或信息内容。与熵不同的是，微分熵可能出现负值，事实上，取决于概率密度函数，它可在$(-\infty,\infty)$范围内变化。

在 8.2.2 节中为离散信源做出的所有其他的定义(包括联合熵、条件熵以及互信息)可以类似地为连续信源做出，只须把任何所涉及到的概率质量函数用相应的概率密度函数代替即可。类似地，熵、条件熵和互信息之间的关系式仍然成立，只须用微分熵代替熵。

高斯信源　一个非常重要的连续信源是高斯信源，它的每个样点都符合高斯分布，且每组 N 个样点符合联合高斯分布。这里，我们给出高斯随机变量(RV)和高斯随机矢量的微分熵。

具有均值 η 和方差 σ^2 的高斯 RV 的概率密度函数为：

$$p(f)=\frac{1}{\sqrt{2\pi}\sigma}e^{-(f-\eta)^2/2\sigma^2},\quad f\in(-\infty,\infty) \tag{8.2.22}$$

可以证明(习题 8.4)它的微分熵是：

$$h_{\text{Gaussian}}=\frac{1}{2}\log_2(2\pi e\sigma^2) \tag{8.2.23}$$

一个均值为 $\boldsymbol{\eta}$，协方差矩阵为$[\mathbf{C}]$的 N 维高斯随机矢量$\mathcal{F}$的联合概率密度函数为：

$$p(\mathbf{f})=\frac{1}{(\sqrt{2\pi})^N\,|\det[\mathbf{C}]|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{f}-\boldsymbol{\eta})^T[\mathbf{C}]^{-1}(\mathbf{f}-\boldsymbol{\eta})\right) \tag{8.2.24}$$

可以证明这个随机矢量的微分熵(或 N 个分量的联合微分熵)是：

$$h_{\text{Gaussian}-N}=\frac{1}{2}\log_2((2\pi e)^N|\det[\mathbf{C}]|)=\frac{1}{2}\log_2\left((2\pi e)^N\prod_n\lambda_n\right) \tag{8.2.25}$$

其中 $\lambda_n(n=1,2,\cdots,N)$是$[\mathbf{C}]$的特征值。

高斯分布的一个重要特征是高斯 RV 在具有相同方差的所有连续 RV 中有最大的熵。类似地，高斯随机矢量在具有相同协方差矩阵的所有随机矢量中有最大的熵。这就使高斯信源最难于编码，因为对给定的失真准则它需要用比其他任何信源都要高的比特率来表示。

---

① 前面基于概率质量函数定义的熵也称为绝对熵。

8.3 信源编码的信息理论

信息论的几个重要结果建立了信源编码的理论基础①。源于信息论之父香农(Shannon)的这些结果建立了实现无损编码和有损编码所需要的最低比特率的界限。严格并全面地介绍这个课题超出了本书的范围。本节的内容是要帮助读者理解信源编码的理论极限,以及学习如何运用它们来指导实际编码技术的设计。为了更深入研究,读者可参考Cover和Thomas[3]以及Berger[1]。

8.3.1 无损编码的界限

标量无损编码 对于用字母表$\mathcal{A}=\{a_1,a_2,\cdots,a_L\}$表示的离散信源$\mathcal{F}$的一个特定实现$\{f_n\}$,标量无损编码是指给每个样点$f_n$分配一个二进制码字$c_n$。这要求预先设计一个码书$C=\{c(a_1),c(a_2),\cdots,c(a_L)\}$,其中,$c(a_i)$是符号$a_i$的码字。于是$f_n$的码字就是$c_n=c(f_n)$。为了使码书是有用的,编码序列必须是惟一可解码的;也就是说,一个码字序列必须对应一个且只有一个可能的信源符号序列。注意,这是比字母表与码书之间的一对一映射更强的要求。令$l(a_i)$表示$c(a_i)$的长度(即比特数),则比特率定义为每个样点的平均比特数②,等于每个码字的平均长度,也就是:

$$R=\sum_{a_i\in\mathcal{A}}p(a_i)l(a_i) \tag{8.3.1}$$

定理 8.1(标量无损编码的界限) 通过为每个样点分配一个码字来表示离散平稳信源$\mathcal{F}$,所需要的最小比特率$R_1(\mathcal{F})$满足:

$$H_1(\mathcal{F})\leqslant\bar{R}_1(\mathcal{F})\leqslant H_1(\mathcal{F})+1 \tag{8.3.2}$$

当信源的概率质量函数是2的幂函数时可达到下限。也就是说,存在一系列整数$\{m_1,m_2,\cdots,m_L\}$使$p(a_i)=2^{-m_i}$。在这种情况下,$l(a_i)=-\log_2 p(a_i)=m_i$。

定理8.1告诉我们离散信源的一阶熵$H_1(\mathcal{F})$决定了信源标量编码的最低比特率的范围。这并不奇怪,因为$H_1(\mathcal{F})$度量信源中的每个新样点所携带的信息内容(以比特/样点表示)。

矢量无损编码 上述定理告诉我们,与信源有关,最小可达到的比特率可以与一阶熵相差1比特/样点。为了进一步减少比特率,可把离散序列中每组相继的N个样点作为一个矢量采样对待,并为$\mathcal{A}^N$中的每个可能的矢量符号分配一个码字。矢量信源的一阶熵是原始信源的N阶熵。如果$\bar{R}^N$表示每个矢量的最小比特数,那么把定理8.1用于上述矢量信源就有:

$$H_N(\mathcal{F})\leqslant\bar{R}^N(\mathcal{F})\leqslant H_N(\mathcal{F})+1 \tag{8.3.3}$$

如果令$R_N=R^N/N$表示每个样点的最小比特数,就得出如下定理。

定理 8.2(矢量无损编码界限) 若通过给每组N个相继样点分配一个码字来表示离散平稳信源$\mathcal{F}$,所需要的最小比特率$\bar{R}_N(\mathcal{F})$满足:

---

① 信源编码是指把样点从信源转成二进制比特流的过程,而信道编码是指进一步处理信源比特,以加入抗传输误码的保护。

② 在本书的其他地方,我们也把比特率定义为每秒比特数(bps)。对于每秒输出f_s个样点并且用R个比特表示每个样点的信源,以bps为单位的比特率是f_sR。

$$H_N(\mathcal{F})/N \leqslant \bar{R}_N(\mathcal{F}) \leqslant H_N(\mathcal{F})/N + 1/N \tag{8.3.4}$$

当 $N \to \infty$ 的极限情况下，由公式(8.2.16)，我们有：

$$\lim_{N \to \infty} \bar{R}_N(\mathcal{F}) = \bar{H}(\mathcal{F}) \tag{8.3.5}$$

定理 8.2 告诉我们将许多样点一起编码可使比特率任意地接近信源熵率。

条件无损编码 我们已经看到借助矢量编码可改进标量编码的效率。另外一种方法是条件编码，也称为基于上下文的编码。对于 M 阶条件编码，当前样点的码字取决于前面 M 个样点所形成的模式。称这种模式为上下文。具体地说，基于给定上下文的输出样点的条件分布，对每种可能的上下文设计分离的码书。如果表示信源的字母表的长度为 L，那么上下文的最大数目，也就是码书的最大数目为 L^M。将定理 8.1 应用于上下文 m 下的条件分布，在这种上下文下的最小比特率满足：

$$H^m_{C,M}(\mathcal{F}) \leqslant \bar{R}^m_{C,M}(\mathcal{F}) \leqslant H^m_{C,M}(\mathcal{F}) + 1 \tag{8.3.6}$$

其中，$H^m_{C,M}(\mathcal{F})$ 表示上下文 m 下的 M 阶条件分布熵。令 p_m 表示上下文 m 的概率，则对于条件阶 M 的平均最小比特率是：

$$\bar{R}_{C,M}(\mathcal{F}) = \sum_m p_m \bar{R}^m_{C,M}(\mathcal{F}) \tag{8.3.7}$$

把公式(8.3.6)代入上式就得到以下结果。

定理 8.3（条件无损编码界限） 用 M 阶条件编码表示离散平稳信源 $\mathcal{F}$，所需要的最小比特率 $\bar{R}_{C,M}(\mathcal{F})$ 满足：

$$H_{C,M}(\mathcal{F}) \leqslant \bar{R}_{C,M}(\mathcal{F}) \leqslant H_{C,M}(\mathcal{F}) + 1 \tag{8.3.8}$$

在 $M \to \infty$ 的极限情况下，我们有：

$$\bar{H}(\mathcal{F}) \leqslant \lim_{M \to \infty} \bar{R}_{C,M}(\mathcal{F}) \leqslant \bar{H}(\mathcal{F}) + 1 \tag{8.3.9}$$

与公式(8.3.2)比较，条件编码可比标量编码达到更低的比特率，因为 $\bar{H}(\mathcal{F}) < H_1(\mathcal{F})$——除非信源是独立恒等分布。然而，因为一次仍然编码一个样点，所以即使条件阶 M 趋于 ∞，上限仍与熵率差 1 比特。N 阶矢量无损编码与第 $(N-1)$ 阶条件编码比较，按照公式(8.2.18)，前者总有相同的或更大的下限。然而，如果 $\frac{1}{N}H_N(\mathcal{F}) - H_{C,N-1}(\mathcal{F}) < (N-1)/N$，矢量编码就有更小的上限。因此，哪个方法更有效取决于实际信源的统计特性。

8.3.2 有损编码的界限*

如果信源是连续的，用有限的比特数准确地表示它是不可能的。回想连续信源的绝对熵率是无穷的。因此，将无损编码界限（公式(8.3.5)）应用于连续信源就会得到一个无穷的最小比特率。为了用有限比特数描述连续信源，必须以某种方法量化信源，以便每个样点或每组样点只可从有限字母表中取符号。这种量化不可避免地会引起原始信源与基于量化表示的重建信源之间的失真。我们可以直接量化信源样点，可以一次量化一个样点（标量量化）或一次量化一组样点（矢量量化），或把样点变换成一些其他的参数并量化所得的参数。然后可把量化后的符号无损编码成二进制码字。在解码器中，首先把码字映射回量化后的符号，再用符号重

---

* 带有星号的节可以跳过或者可留待进一步的研究。

建每个样点。

一般地，可将有损编码过程认为是把每个原始信源$\mathcal{F}$的 N 个样点组成的输入矢量 $\mathbf{f}=\{f_1,f_2,\cdots,f_N\}$映射到一个量化矢量 $\mathbf{g}=Q(\mathbf{f})$。矢量 $\mathbf{g}$ 必须属于预先设计的有限长度 L 的重建码书，$C=\{\mathbf{g}_1,\mathbf{g}_2,\cdots,\mathbf{g}_L\}$。我们将假设 $\mathbf{f}$ 和 $\mathbf{g}$ 的每个分量只可从相同的支撑区$\mathcal{B}$中取值。采用固定长度编码，每个量化矢量用 $\log_2(L)$个比特表示，则编码序列的比特率(比特/样点)是：

$$R_N = \frac{1}{N}\log_2(L) \tag{8.3.10}$$

给定一个预先设计的重建码书，通过对码字的索引应用可变长编码，由公式(8.3.10)所得的比特率可以进一步减少。然而，在下面的讨论中，我们假设按照公式(8.3.10)来确定比特率。

为了满足$\mathcal{F}$和$\mathcal{G}$之间的给定失真准则，所需要的码书长度 L 取决于编码方案(即码书是如何设计的)。从直觉上来说，增加矢量长度 N 可减少比特率，特别是当信源不是独立恒等分布时。率失真理论定义了当 N 趋于∞时，达到给定失真所需要的比特率的下限。

注意，有损编码可应用于连续的和离散的两种信源。回想对于离散信源，无损(零失真)编码所需要的最小比特率是受信源熵率限制的。为了达到更低的码率，必须把样点量化到较小的字母表，因此就会产生一定程度的失真。以下讨论适用于连续的和离散的信源。

失真测度　到目前为止，我们已使用了术语"失真"一词，但没有正式定义它。为了度量原始信源$\mathcal{F}$和重建信源$\mathcal{G}$之间的失真，我们首先定义两个标量 f 和 g 之间的失真测度 $d_1(f,g)$。如果 f 和 g 都是实数，最常用的失真测度是平方误差 $d_1(f,g)=(f-g)^2$ 和绝对误差 $d_1(f,g)=|f-g|$。这样，原始矢量 $\mathbf{f}$ 和重建矢量 $\mathbf{g}$ 之间的每个样点的平均失真是：

$$d_N(\mathbf{f},\mathbf{g}) = \frac{1}{N}\sum_n d_1(f_n,g_n) \tag{8.3.11}$$

这个失真是为给定的矢量和它的重建矢量定义的。为了评估所有可能的输入矢量的平均性能，当矢量长度为 N 时，我们定义如下的$\mathcal{F}$和$\mathcal{G}$之间的期望失真：

$$E\{d_N(\mathcal{F},\mathcal{G})\} = \int_{\mathbf{f}\in\mathcal{B}^N}\int_{\mathbf{g}\in\mathcal{B}^N} p_N(\mathbf{f})q_N(\mathbf{g}\mid\mathbf{f})d_N(\mathbf{f},\mathbf{g})d\mathbf{g}d\mathbf{f} \tag{8.3.12}$$

其中，$p_N(\mathbf{f})$表示$\mathcal{F}$中的 N 个相继样点的联合概率密度函数，$q_N(\mathbf{g}|\mathbf{f})$表示 $\mathbf{f}$ 被重建为 $\mathbf{g}$ 的条件概率，而$\mathcal{B}^N$ 表示$\mathcal{B}$的 N 重笛卡儿积。注意，$q_N(\mathbf{g}|\mathbf{f})$从根本上表征了编码方案，尽管在实际中是采用 $\mathbf{f}$ 和 $\mathbf{g}$ 之间的确定性的映射。当 $d_1(f,g)$是平方误差时，上述测度称为均方误差(MSE)；MSE 得到广泛应用，主要是因为它在数学上易于处理。

以上，我们假设信源是连续的，函数 $p_N(\cdot)$，$q_N(\cdot)$表示概率密度函数。如果信源是离散的，应该用$\mathcal{A}^N$ 上的求和代替$\mathcal{B}^N$ 上的积分，而 $p_N(\cdot)$，$q_N(\cdot)$则表示相应的概率质量函数。

率失真界限　有损编码器的性能是通过运算率失真(RD)函数度量的，表示为 $R(D)$，对于给定的信源特征，它描述达到给定失真 D 所需要的比特率 R。率失真定理(定理 8.4)建立了当矢量长度 N 趋于∞时，在所有可能的编码器中所需要的最小码率的界限。我们称最小码率与失真的关系函数为率失真界限，用 $\bar{R}(D)$表示。令 $R_N(D;q_N(\mathbf{g}|\mathbf{f}))$表示一个用 $q_N(\mathbf{g}|\mathbf{f})$描述的编码方案为达到失真 D 所需要的比特率。则 RD 界限定义为：

$$\bar{R}(D) = \lim_{N\to\infty}\min_{q_N(\mathbf{g}|\mathbf{f})\in Q_{D,N}} R_N(D;q_N(\mathbf{g}\mid\mathbf{f})) \tag{8.3.13}$$

其中

$$Q_{D,N} = \{q_N(\mathbf{g} \mid \mathbf{f}) : E\{d_N(\mathcal{F},\mathcal{G})\} \leqslant D\} \tag{8.3.14}$$

是条件概率集(即编码方案),它满足所规定的失真约束。

定理 8.4(有损编码界限)　在失真 D 下编码平稳信源$\mathcal{F}$的 RD 界限由下式给出[①]:

$$\bar{R}(D) = \lim_{N\to\infty} \min_{q_N(\mathbf{g}|\mathbf{f})\in Q_{D,N}} \frac{1}{N} I_N(\mathcal{F};\mathcal{G}) \tag{8.3.15}$$

当信源是独立恒等分布(i.i.d.)时,RD 界限变成:

$$\bar{R}(D) = \min_{q_1(\mathbf{g}|\mathbf{f})\in Q_{D,1}} I_1(\mathcal{F};\mathcal{G}) \tag{8.3.16}$$

回想 $I_N(\mathcal{F};\mathcal{G})$描述$\mathcal{G}$的 N 个样点关于相应的$\mathcal{F}$的 N 个样点所提供的信息量。因此,不用奇怪 - $I_N(\mathcal{F};\mathcal{G})$可以确定当 $N\to\infty$ 且失真为 D 时为表示$\mathcal{F}$所需的比特数。

信息论的一个重要结果是,即使当信源是 i.i.d. 时,仍可以通过将许多样点一起编码来减少比特率 R。实际上,只有当矢量长度 N 趋于∞时,RD 界限才是可达到的。

一般地,定理 8.4 中的界限对于任意的信源是很难计算的。下面的定理描述了一个更有用的界限。

定理 8.5　在 MSE 失真准则下,任何平稳信源$\mathcal{F}$的 RD 界限满足:

$$\bar{R}_L(D) \leqslant \bar{R}(D) \leqslant \bar{R}_G(D) \tag{8.3.17}$$

其中,$\bar{R}_G(D)$是与$\mathcal{F}$有相同方差的高斯信源的 RD 界限,而 $\bar{R}_L(D)$称为香农下限,表示为:

$$\bar{R}_L(D) = \bar{h}(\mathcal{F}) - \frac{1}{2}\log_2 2\pi eD = \frac{1}{2}\log_2 \frac{Q(\mathcal{F})}{D} \tag{8.3.18}$$

其中,$\bar{h}(\mathcal{F})$是$\mathcal{F}$的熵率,而 $Q(\mathcal{F}) = (1/2\pi e)2^{2\bar{h}(\mathcal{F})}$称为$\mathcal{F}$的熵幂。对于高斯信源,$\bar{R}_L(D) = \bar{R}_G(D)$。

定理 8.5 告诉我们,在具有相同方差的所有信源中,为满足相同的失真准则,高斯信源需要的比特率最高。实际上,对于任何信源,满足 RD 界限的最优编码方案应使量化误差序列是一个方差为 D,微分熵为$\frac{1}{2}\log_2 2\pi eD$ 的 i.i.d. 高斯信源。因此,公式(8.3.18)中给定的下限等于原始信源微分熵和量化误差的微分熵之间的差。

8.3.3　高斯信源率失真的界限*

对于大多数的信源,很难求出 RD 界限封闭形式的解。各种类型的高斯信源是一组例外,当失真准则为 MSE 时,已经找到了其封闭形式的解。下面我们概括这些结果。

i.i.d. 高斯信源　具有方差 σ^2 的 i.i.d. 高斯信源的 RD 界限是:

$$\bar{R}(D) = \begin{cases} \frac{1}{2}\log_2 \frac{\sigma^2}{D}, & 0 \leqslant D < \sigma^2 \\ 0 & D \geqslant \sigma^2 \end{cases} \tag{8.3.19}$$

或者,可写为:

$$\bar{D}(R) = \sigma^2 2^{-2R} \tag{8.3.20}$$

---

① 在信息论中,信源$\mathcal{F}$的 RD 函数定义为等式(8.3.15)的右边。这里我们称它为 RD 界限。香农信源编码定理指出存在一种编码方案,使达到失真 D 所需要的比特率接近 $\bar{R}(D)$[1,3]。

* 带有星号的节可以跳过或者可留待进一步的研究。

具有独立分量的 i.i.d. 矢量高斯信源 考虑一个 i.i.d. 矢量高斯信源，其中每个矢量采样由方差为 $\sigma_n^2, n=1,2,\cdots,N$ 的 N 个独立高斯 RV 组成。这种信源的 RD 界限是：

$$\begin{cases} R(\alpha) = \frac{1}{N}\sum_n \max\left\{0, \frac{1}{2}\log_2 \frac{\sigma_n^2}{\alpha}\right\} \\ D(\alpha) = \frac{1}{N}\sum_n \min\{\alpha, \sigma_n^2\} \end{cases} \tag{8.3.21}$$

其中，α 是一个中间参数。令 α 遍历范围 $(0, \max\{\sigma_n^2\})$ 就可获得合理的失真区域的 RD 界限。

当 D 足够小，即 $D \leqslant \min\{\sigma_n^2\}$ 时，前面的公式简化为：

$$\bar{R}(D) = \frac{1}{2}\log_2 \frac{(\prod_n \sigma_n^2)^{1/N}}{D} \tag{8.3.22}$$

或

$$\bar{D}(R) = \left(\prod_n \sigma_n^2\right)^{1/N} 2^{-2R} \tag{8.3.23}$$

对一个矢量采样中的每个分量进行独立编码可达到这个 RD 界限，用于分量 n 的比特数是 $R_n = \max\left\{0, \frac{1}{2}\log_2(\sigma_n^2/\alpha)\right\}$。对于 $\sigma^2 \leqslant \alpha$ 的分量总可以用零来表示，从而 $R_n = 0$。对于其他分量，比特的分配应使它们都有相同的失真 $D_n = \alpha$。这个结果可以通过在 N 个分量中间最优地分配总的比特率 NR 使平均失真 D 为最小来达到。实质上，它要求选择比特率 R_n，使得不同分量的率失真函数的斜率 $\partial D_n(R_n)/\partial R_n$ 在这些比特率上都相等。

具有相关分量的 i.i.d. 矢量高斯信源 令[C]表示每个矢量采样的 N 个分量的协方差矩阵，而 $\lambda_n (n=1,2,\cdots,N)$ 表示[C]的特征值。这种信源的 RD 界限是：

$$\begin{cases} R(\alpha) = \frac{1}{N}\sum_n \max\left\{0, \frac{1}{2}\log_2 \frac{\lambda_n}{\alpha}\right\} \\ D(\alpha) = \frac{1}{N}\sum_n \min\{\alpha, \lambda_n\} \end{cases} \tag{8.3.24}$$

其中，$\alpha \in (0, \max\{\lambda_n\})$ 是一个中间参数。

当 D 足够小，即 $D \leqslant \min\{\lambda_n\}$ 时，前面的公式简化为：

$$\bar{R}(D) = \frac{1}{2}\log_2 \frac{\left(\prod_n \lambda_n\right)^{1/N}}{D} = \frac{1}{2}\log_2 \frac{|\det[\mathbf{C}]|^{1/N}}{D} \tag{8.3.25}$$

或

$$\bar{D}(R) = \left(\prod_n \lambda_n\right)^{1/N} 2^{-2R} = |\det[\mathbf{C}]|^{1/N} 2^{-2R} \tag{8.3.26}$$

用一个由[C]的特征矢量组成的变换矩阵来变换原始矢量，此时变换后的矢量具有独立分量，然后对它应用公式(8.3.21)中的 RD 界限，就得到上述结果。

一般高斯信源 均值为 η 的任意平稳高斯信源可由它的自协方差函数 $C(k) = E\{(\mathcal{F}_n - \eta)(\mathcal{F}_{n+k} - \eta)\} (k = 0, \pm 1, \cdots,)$ 或它的功率谱 $S(e^{j\omega})$① 来描述。这种信源的 RD 界限是在公式(8.3.24)中所给出的 RD 界限当 $N \to \infty$ 时的极限。结果是：

① $S(e^{j\omega})$是 $C(k)$的离散时间傅里叶变换。

$$\begin{cases} R(\alpha) = \dfrac{1}{2\pi}\displaystyle\int_{-\pi}^{\pi} \max\left\{0, \dfrac{1}{2}\log_2 \dfrac{S(e^{j\omega})}{\alpha}\right\} d\omega \\ D(\alpha) = \dfrac{1}{2\pi}\displaystyle\int_{-\pi}^{\pi} \min\{\alpha, S(e^{j\omega})\} d\omega \end{cases} \tag{8.3.27}$$

当 D 足够小,使得 $D > S(e^{j\omega})$的 ω 区域可忽略不计时,我们有:

$$R(D) = \frac{1}{4\pi}\int_{-\pi}^{\pi} \log_2 \frac{S(e^{j\omega})}{D} d\omega \tag{8.3.28}$$

公式(8.3.27)告诉我们为了表示失真为 α、功率谱为 $S(e^{j\omega})$的信源,$S(e^{j\omega}) < \alpha$ 的频率区域不必编码,而其余区域的编码应该使这个区域上的误差信号具有与 α 相等的功率。这个比特分配的过程通常称为逆向注水。

在具有 $C(k) = \sigma^2 |\rho|^k$ 的一阶高斯 - 马尔可夫过程的特殊情况下,我们有:

$$R(D) = \frac{1}{2}\log_2 \frac{(1-\rho^2)\sigma^2}{D} \tag{8.3.29}$$

或

$$D(R) = (1-\rho^2)\sigma^2 2^{-2R} \tag{8.3.30}$$

8.4 二进制编码

二进制编码是用二进制比特序列(称为码字)表示有限字母表信源中每个可能符号的过程。所有可能符号的码字形成码书。一个符号可以对应一个或几个原始的或量化后的像素值或模型参数。因为从符号到码字的映射是一一对应的,因此这个过程也称为无损编码。

对于一个有用的码,它应该满足以下属性:(1)它应该可惟一解码,这就意味着在码字和符号之间有一对一映射的关系;(2)码应该是即时可解码的,这意味着如果一组比特与码字相匹配,那么可立即解码这组比特,而不需检查编码序列中的后继比特(见图 8.2)。这第二个属性要求任何码字的前缀都不是另一个有效的码字;这种码称为前缀码。尽管即时可解码性是比惟一性更强的要求,而且允许快速解码,但它不限制编码效率。可以证明,对于同一信源,在所有惟一可解码的码中前缀编码可产生最小比特率[4]。所有实际编码方法都产生前缀码。

很明显,最简单的二进制码是所有可能符号的固定长度的二进制表示。如果符号数是 L,那么比特率就是$\lceil \log_2 L \rceil$比特/符号。由 8.3.1 节我们知道,任何码书的最低可能比特率是信源的熵率。除非信源是均匀分布的,否则固定长度编码方案将是效率很低的,因为比特率比熵率高得多。为了降低比特率,需要可变长编码(VLC),它分配一个较短的码字给一个较高概率的符号,所以平均比特率低。因为适当设计的可变长编码器的比特率可接近信源的熵,所以可变长编码也称为熵编码。

有三种流行的可变长编码方法。哈夫曼(Huffman)编码把固定数目的符号转成可变长的码字;LZW 方法把可变数目的符号转成固定长度的码字;而算术编码把可变数目的符号转成可变长度的码字。哈夫曼和算术方法是基于概率模型的,且都可逐渐地达到熵界限。算术编码方法更容易达到渐进性能;且容易适应信号统计特性的变化,但它比哈夫曼编码更复杂。LZW 方法[12,10]不要求了解信号的统计特性,因此是普遍适用的,但它比其他两种方法的效率低。哈夫曼和算术编码已经用于各种视频编码标准中。在以下两小节中将描述这两种方法。

码书1
(前缀码)

符号	码字
a_1	"0"
a_2	"10"
a_3	"110"
a_4	"111"

码书2
(非前缀码)

符号	码字
a_1	"0"
a_2	"01"
a_3	"100"
a_4	"011"

比特流： 0 0 1 1 0 1 0 1 1 0 1 0 0

基于码书1解码的串(可即时解码)： 0|0|1 1 0|1 0|1 1 0|1 0|0 → $a_1\ a_1\ a_3\ a_2\ a_3\ a_2\ a_1$

基于码书2解码的串(必须考虑未来的比特才能解码)： 0|0 1 1 0 1|0 1 1|0 1 0 0 → $a_1\ a_4\ a_2\ a_4\ a_1\ a_3$

图 8.2 对于左边的码书(前缀码)，给定的比特流可以在检测到一个完整的码字时即时解码。对于右边的码书(非前缀码)，人们不知道比特流中的第一个"0"是对应码字"0"，还是对应第二个码字或第四个码字中的第一个比特。必须检查其他的比特以做出判定。因为第二个比特仍是"0"，然而没有起始是两个"0"的码字，所以可知第一个"0"必须对应码字"0"

8.4.1 哈夫曼编码

哈夫曼编码是具有字母表$\mathcal{A}=\{a_1,a_2,\cdots,a_L\}$和概率质量函数$p(a_l)$的离散信源无损编码最流行的方法。它设计针对所有可能符号的码书，使得出现较频繁的符号被分配较短的码字。设计哈夫曼编码码书的基本过程如下：

步骤 1：以降序排列符号概率$p(a_l)(l=1,2,\cdots,L)$，并把它们作为树的叶节点。

步骤 2：当存在一个以上的节点时：

(a) 找到具有最小概率的两个节点，并任意地分配 1 和 0 给这两个节点。

(b) 合并这两个节点形成一个新的节点，它的概率是这两个合并节点的概率之和。返回步骤 1。

步骤 3：对于每个符号，通过跟踪从对应的叶节点到树的顶端所分配的比特来确定它的码字。叶节点的比特是该码字的最后一个比特。

例 8.1 (标量哈夫曼编码)

一个哈夫曼编码的例子如图 8.3 所示，在图中信源由 4 个符号组成。左边两列给出了符号和它们的概率。右边两列给出这些符号所得的码字和码字的长度。也给出了平均比特率 R 和信源的一阶熵 H_1。实际上我们可以看到 $H_1<R<H_1+1$。

哈夫曼编码显著的缺点是：当应用于各个样点时，一个样点必须至少用一个比特。为了进一步减少比特率，可把每组 N 个样点作为一个实体对待，并给每组一个码字。这就导致矢量哈夫曼编码，它是矢量无损编码的一个特殊情况。这在下一个例子中给出。

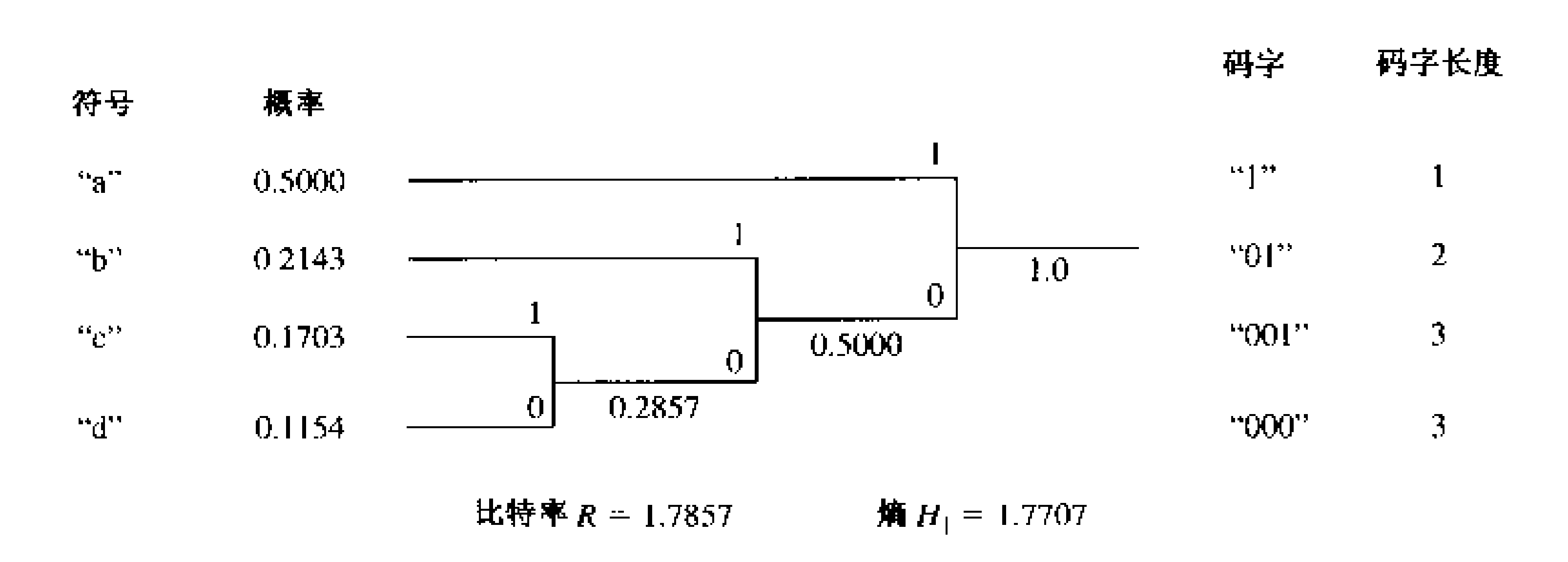


图 8.3　标量哈夫曼编码的例子(见例 8.1)

例 8.2　(矢量哈夫曼编码)

考虑与例 8.1 一样的字母表和概率质量函数。假设我们进一步知道给定前面一个样点 $\mathcal{F}_{n-1}$ 时,样点 $\mathcal{F}_n$ 的条件分布由如下矩阵所描述:

$$[\mathbf{Q}] = \begin{bmatrix} 0.6250 & 0.3750 & 0.3750 & 0.3750 \\ 0.1875 & 0.3125 & 0.1875 & 0.1875 \\ 0.1250 & 0.1875 & 0.3125 & 0.1250 \\ 0.0625 & 0.1250 & 0.1250 & 0.3125 \end{bmatrix} \tag{8.4.1}$$

其中,第 i 行第 j 列的元素确定条件概率 $q(i|j)$,这是在 $\mathcal{F}_{n-1}$ 等于第 j 个符号的条件下 $\mathcal{F}_n$ 取第 i 个符号的概率。我们可确定每两个样点的联合概率质量函数为:

$$p(f_{n-1}, f_n) = p(f_{n-1})q(f_n \mid f_{n-1}) \tag{8.4.2}$$

把它应用于所有可能的两个符号的组合,我们得到所有可能的二维矢量符号概率,它在图 8.4 的第一列中给出。然后,我们可以对这个新的信源应用哈夫曼编码;图 8.4 中也给出了所得到的码书。在这种情况下,每个矢量采样的比特率是 $R^2 = 3.5003$,而每个样点的比特率是 $R_2 = R^2/2 = 1.75015$。另一方面,$H_2/2 = 1.7314$。确实,我们有 $\frac{1}{2}H_2 < R_2 < \frac{1}{2}H_2 + 1/2$。与例 8.1 中的结果比较可以看出,通过矢量编码使比特率下降。

也可以把两个样点一起编码改为条件哈夫曼编码,它根据前面样点所取的符号使用不同的码书。如下例所示。

例 8.3　(条件哈夫曼编码)

继续前一个例子的情况,但我们现在为由前一个样点组成的每一种可能的上下文分别设计一个码书。在这种情况下,只有 4 种可能的上下文。基于公式(8.4.1)中给出的矩阵的第二列的条件概率,图 8.5 示出了上下文"b" (即前一个符号是"b")的码书设计。用同样的步骤,可为另外三种可能的上下文设计码书。对四种上下文"a","b","c"和"d"所得到的比特率是 1.5625, 1.9375,1.9375 和 1.9375,所有可能上下文的平均比特率是 $R_{C,1} = 1.7500$。另一方面,4 种上下文的条件熵分别是:1.5016,1.8829,1.8829 和 1.8829,平均条件熵是 $H_{C,1} = 1.6922$。正如所期望的,$H_{C,1} < R_{C,1} < H_{C,1} + 1$。在这个特殊例子中,$R_{C,1} < R_2$,尽管这可能不是一般情况。

如果信源是马尔可夫 - 1 过程,那么信源的熵率就等于它的一阶条件熵。因此,$H_{C,1}$ 是矢量编码所能达到的最低比特率,尽管此时矢量的长度为无穷大。

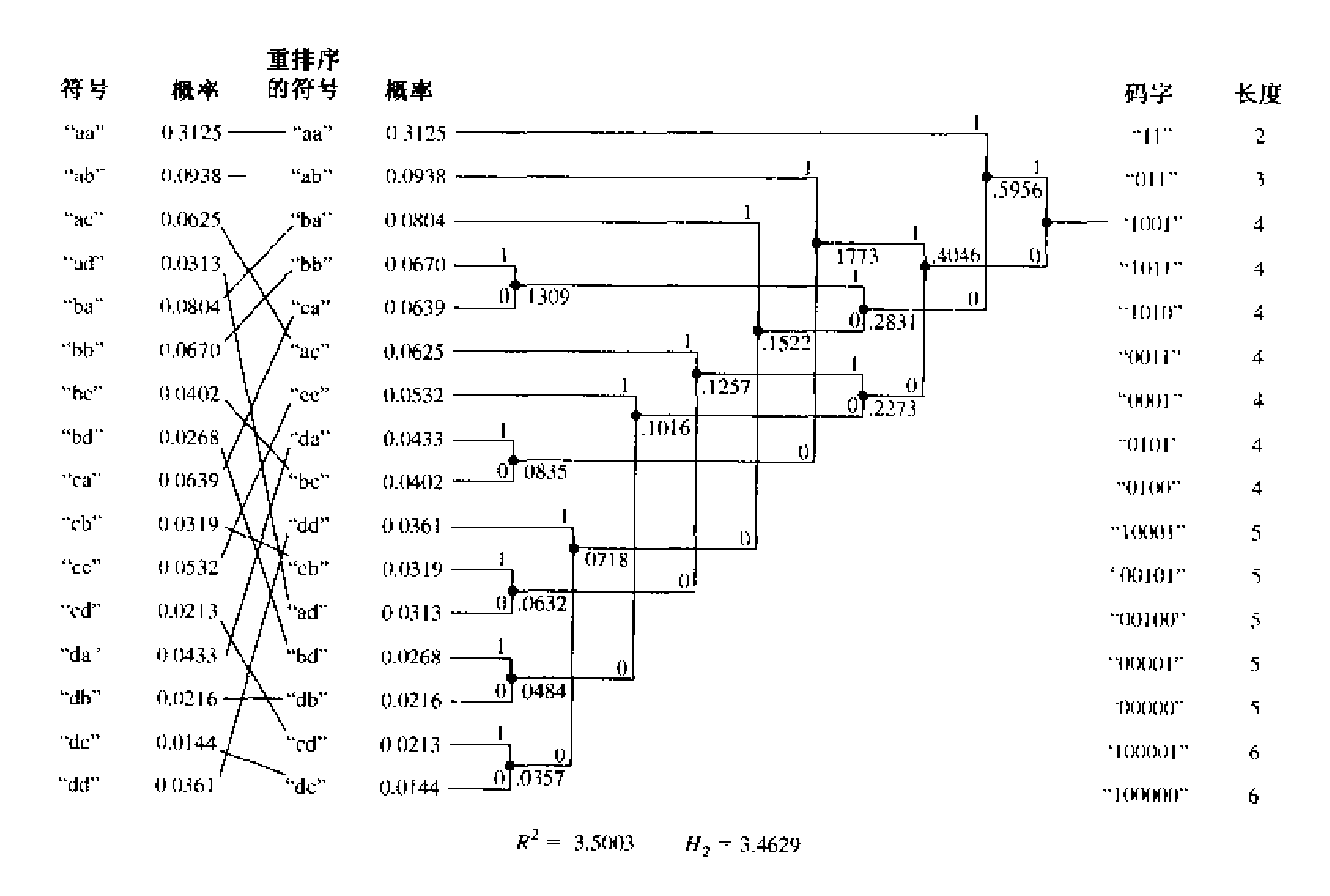


图 8.4 矢量哈夫曼编码的例子(见例 8.2)

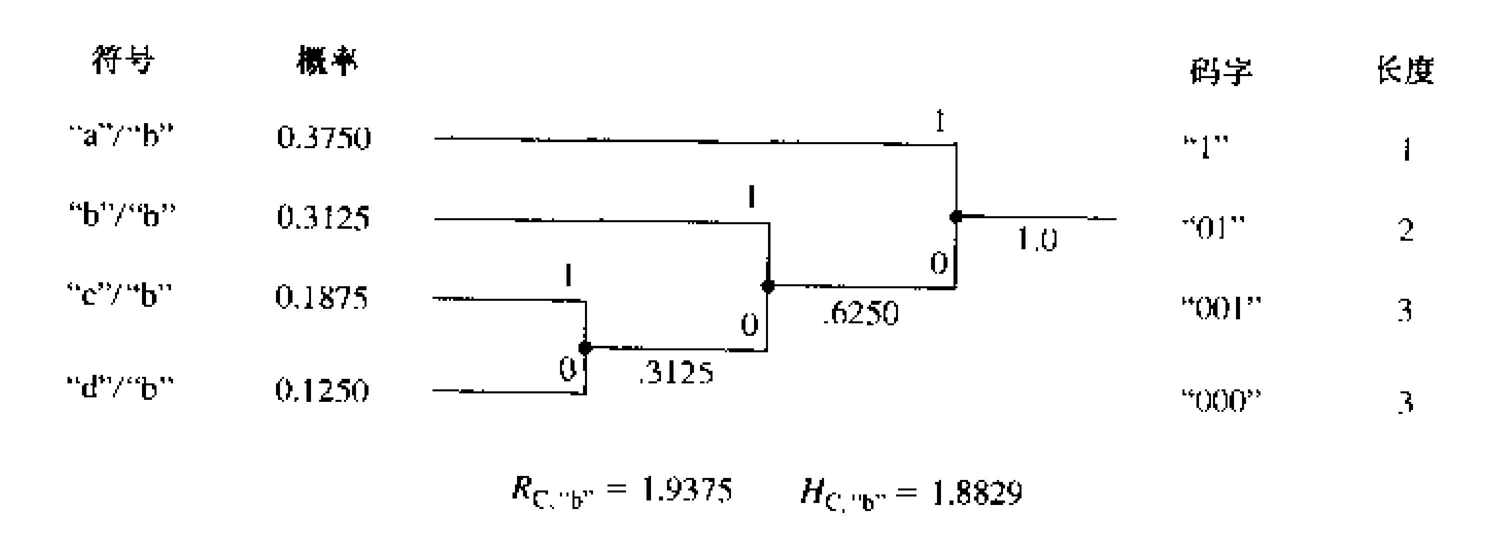


图 8.5 条件哈夫曼编码的例子,此时条件上下文为"b"(见例 8.3)

8.4.2 算术编码

哈夫曼编码的缺点是它不能接近熵界限,除非许多样点一起编码。这是因为每个样点(或样点组)至少用一个比特。因此,即使对一个很小的字母表,比特率在标量编码情况下不可能低于 1 比特/样点,或在矢量编码情况下不可能低于 1 比特/矢量采样。避免这种问题的一种方法是把可变数目的样点转换成可变长的码字。这种方法称为算术编码。对于一般的信号,它可以比哈夫曼编码更接近熵界限。

算术编码的思想是用 0 到 1 的线段上的一个区间表示一个符号序列,其长度等于这个序列的概率。因为所有序列概率和为 1,所以对应于所有可能序列的区间将填满整个线段。一个序列的编码比特实际上就变成对应该序列的区间中任何一点的二进制表示。在确定区间及其二进制表示前不需等到整个序列出现,我们可以从基于第一个符号确定的初始区间开始,然后在每一个新的符号出现后递归地划分前一个区间。为了确定区间,区间的上下限以二进制

形式表示。每当下限的最高有效位(MSB)与上限的最高有效位一样时,就移出这个比特。在信源序列结束时,所有被收集的比特就是在相应于该序列的区间中一个中间点的二进制表示。序列出现的可能性越大,区间就越长,确定该区间所需要的比特数就越少。

令 $a_l(l=1,2,\cdots,L)$表示信源的 L 个可能符号,符号 a_l 的概率为p_l,且 $q_l=\sum_{k=1}^{l}p_l$ 累计直到第 l 个符号的概率。令 d_n, l_n 和 u_n 分别表示第 n 步区间的长度、下限和上限,且 $l_0=0$, $u_0=1$, $d_0=1$。依据收到的信源序列的第 n 个符号,如果符号是 a_l,则下限和上限用下式计算:

$$d_n = d_{n-1} * p_l; \quad l_n = l_{n-1} + d_{n-1} * q_{l-1}; \quad u_n = l_n + d_n \tag{8.4.3}$$

令 l_n 和 u_n 的二进制表示分别是$(b_1^l, b_2^l, \cdots, b_K^l)$和$(b_1^u, b_2^u, \cdots, b_K^u)$①。若前几个 b_k^l 和 b_k^u 是相同的,那么就把它们移出,否则保留所有的比特。然后继续进行算法,检验下一个符号。

例 8.4　图 8.6(a)示出了算术编码的一个例子。信源字母表由"a","b"和"c"3 个符号组成,概率分布为 $p(\text{"a"})=1/2$, $p(\text{"b"})=1/4$, $p(\text{"c"})=1/4$。图 8.6(a)的第一列给出对应这些符号的区间。第一个信源符号是"a",它对应第一列中第一个区间。上下限分别是 $l_1=0=(0000...)$ 和 $u_1=1/2=(1000...)$。因为 l_1 和 u_1 的 MSB 是不同的,所以不移出比特。下一个信源符号是"b",因此,就在第二列中取第二个区间。由公式(8.4.3),这个区间的界是 $l_2=1/4=(01000...)$和 $u_2=3/8=(011000...)$。它们的前两个比特是相同的,因此在编码比特流中就可移出"01"。按照这个程序,可确定所有后面的信源符号的编码比特。

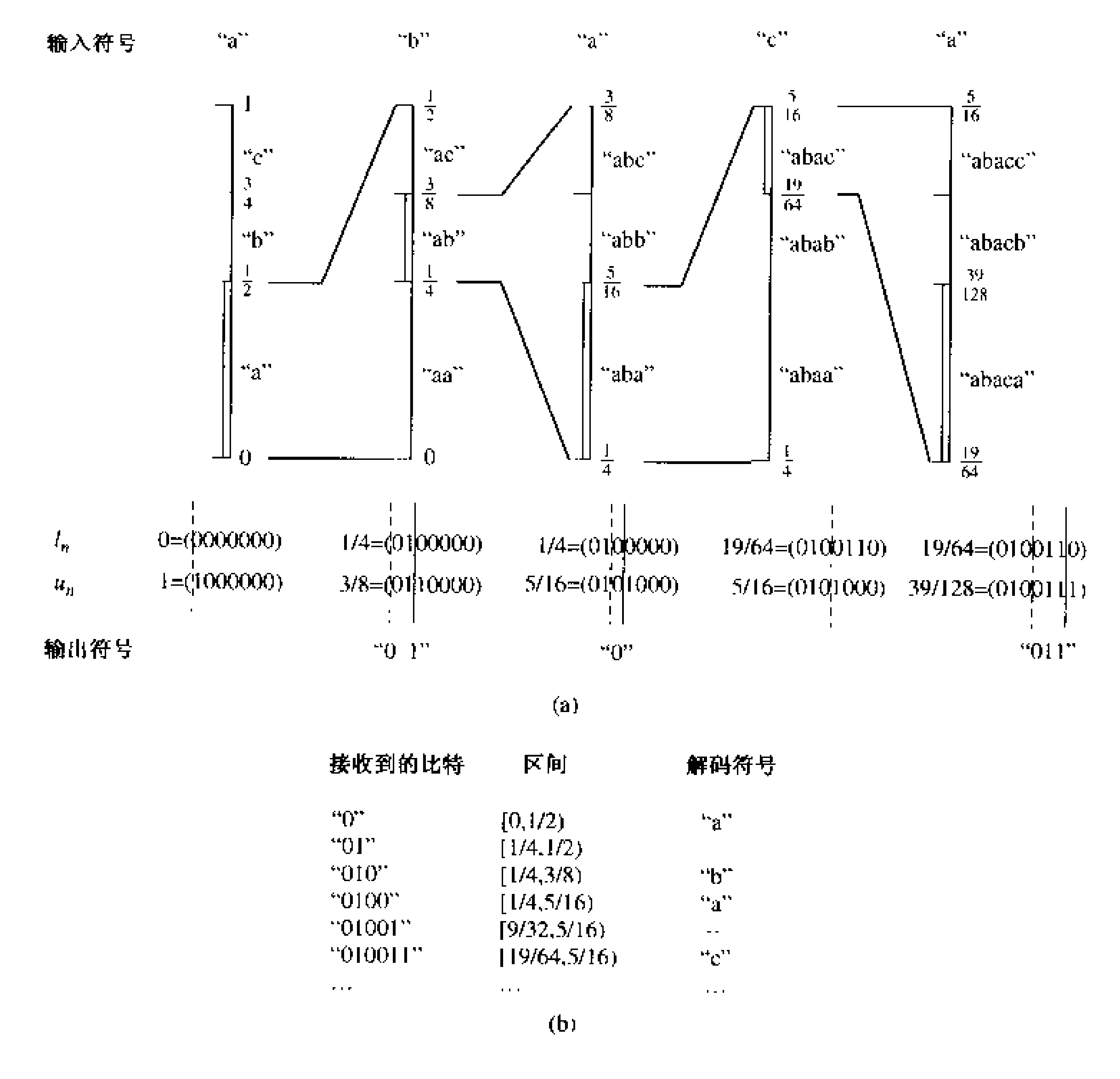
为了解码接收到的比特流,要确定对应接收到的比特的区间的上下限。如果该区间所落入的范围只与一个特殊的信源符号有关,就解码这个符号。否则,就检查其他的比特。

例 8.4(续)　解码过程如图 8.6(b)所示。第一个收到的比特是"0",都以 0 作为最高有效位的可能的下限和上限是 $l=(0000...)=0$ 和 $u=(0111...)=(1000...)=\frac{1}{2}$。从图8.6(a)的第一列可知,这个区间对应"a",因此,可输出"a"作为解码符号。由于下一个接收的比特是"1",都以"01"作为前两个比特的可能的下限和上限是 $l=(0100...)=\frac{1}{4}$ 和 $u=(0111...)=(1000...)=\frac{1}{2}$。从图 8.6(a)的第二列可知,区间$\left(\frac{1}{4},\frac{1}{2}\right)$可以来自"ab"或"ac"。因此,在这一步不能解码出符号。由于第三个比特是"0",下限和上限是 $l=(010000...)=\frac{1}{4}$ 和 $u=(010111...)=(011000...)=\frac{3}{8}$。从图 8.6(a)中的第二列可知,这个区间对应"ab"。因此在这一步我们可以输出"b"。继续这个过程,就可解码出所给比特流的所有符号。

上面描述的算术编码只是概念上的,它假设区间界限的计算是无限精确的。当处理较多的信源符号时,对应输入序列的区间愈加变小。为了处理一个合理长度的序列,需要一个具有极高精度的计算机来确定界限值。幸运的是,在编码和解码过程中不需要记录绝对的界限值。一旦移出下限和上限的公共比特,就可把前面确定的区间重新定标为长度 1。这种实现称为有限精度算术编码。把用于下限和上限计算的小数算术换成更简单的整数实现也已经开发出来。对实际的编解码算法的全面描述超出了本书范围;读者可参考 Witten, Neal 和 Cleary 的优秀教程[11]。在 Sayood[9]中也可找到哈夫曼和算术编码的全面介绍。

---

① 一个 0 和 1 之间的数 f 的二进制表示由$(b_1, b_2, \cdots, b_K)$给出,如果 $f=\sum_{k=1}^{K}b_k2^{-k}$。



接收到的比特	区间	解码符号
"0"	[0,1/2)	"a"
"01"	[1/4,1/2)	
"010"	[1/4,3/8)	"b"
"0100"	[1/4,5/16)	"a"
"01001"	[9/32,5/16)	--
"010011"	[19/64,5/16)	"c"
...	...	...

(b)

图 8.6 算术编码的例子:(a)对信源序列"abaca..."进行编码。(b)解码比特流"010011..."

算术编码的比特率由下式限制:

$$H_N(\mathcal{F})/N \leq R \leq H_N(\mathcal{F})/N + 2/N \tag{8.4.4}$$

其中,N 是编码序列中的符号数而 $H_N(\mathcal{F})$ 是序列的 N 阶熵。回想如果用矢量哈夫曼编码来编码整个长为 N 的序列,比特率将受到公式(8.3.4)的限制。因此,当 N 足够大时,两种方法都可趋近于熵率。然而,用哈夫曼编码,就必须为所有可能的长度为 N 的序列设计和存储码书,这样做的复杂度随 N 呈指数增长。这限制了实际可用的序列长度。用算术编码则不需预先为每个可能的信源序列设计码书。而是每当所确定区间的下限和上限有公共最高有效位时,就可连续地得到比特。编码序列的长度可以与信源的长度一样长。因此,实际上,算术编码能够接近熵率。

算术编码的另一个优点是可简单地通过更新符号概率表来适应信源统计特性的变化。通过对不同的上下文用不同的概率表也可以容易地实现条件编码。对于哈夫曼编码,则不得不基于更新的概率表重新设计码书,或对不同的条件陈述设计多个码书。

由于较高的编码效率和易于自适应,只要所涉及的计算是可接受的,算术编码比哈夫曼编码是一种更好的选择。

8.5 标量量化

8.5.1 基本原理

最基本的有损编码方法是标量量化(SQ),它把信源信号中的每个样点量化成预先设计的重建码书中的重建值之一。一般来讲,原始信源可以是连续的或离散的。这里我们仅考虑第一种情况,即原始样点可以取实线上的连续支撑区$\mathcal{B}$内的任何值。这个支撑区被分成许多区域$\mathcal{B}_l(l=1,2,\cdots,L)$,使得在同一个区域$\mathcal{B}_l$内的值映射到同一个重建值$g_l$上。当信源是离散的,但所有可能的符号可沿实线排列时,我们可以把每个样点作为一个假想的连续信源的输出来对待;这个假想信源在一个连续范围内只产生某些离散值。用这种方法,下面关于连续信源所要讨论的理论和技术也可应用于离散信源。

一个量化器是用重建值的数目L、分界值$b_l(l=0,1,\cdots,L)$和重建值$g_l(l=1,2,\cdots,L)$来描述的。分界值也可以用分割区域$\mathcal{B}_l=[b_{l-1},b_l)$描述①。令$\mathcal{L}=\{1,2,\cdots,L\}$,则量化函数可以描述为:

$$Q(f)=g_l \quad \text{如果} \quad f\in\mathcal{B}_l, l\in\mathcal{L}. \tag{8.5.1}$$

在图 8.7 中图示了这个函数。在数字计算机中,重建值g_l可以简单地用整数索引l确定,它在固定长度二进制表示中需要$R=\lceil\log_2 L\rceil$比特。这里,符号$\lceil x\rceil$表示x的上限整数,也就是等于或大于x的最小整数。

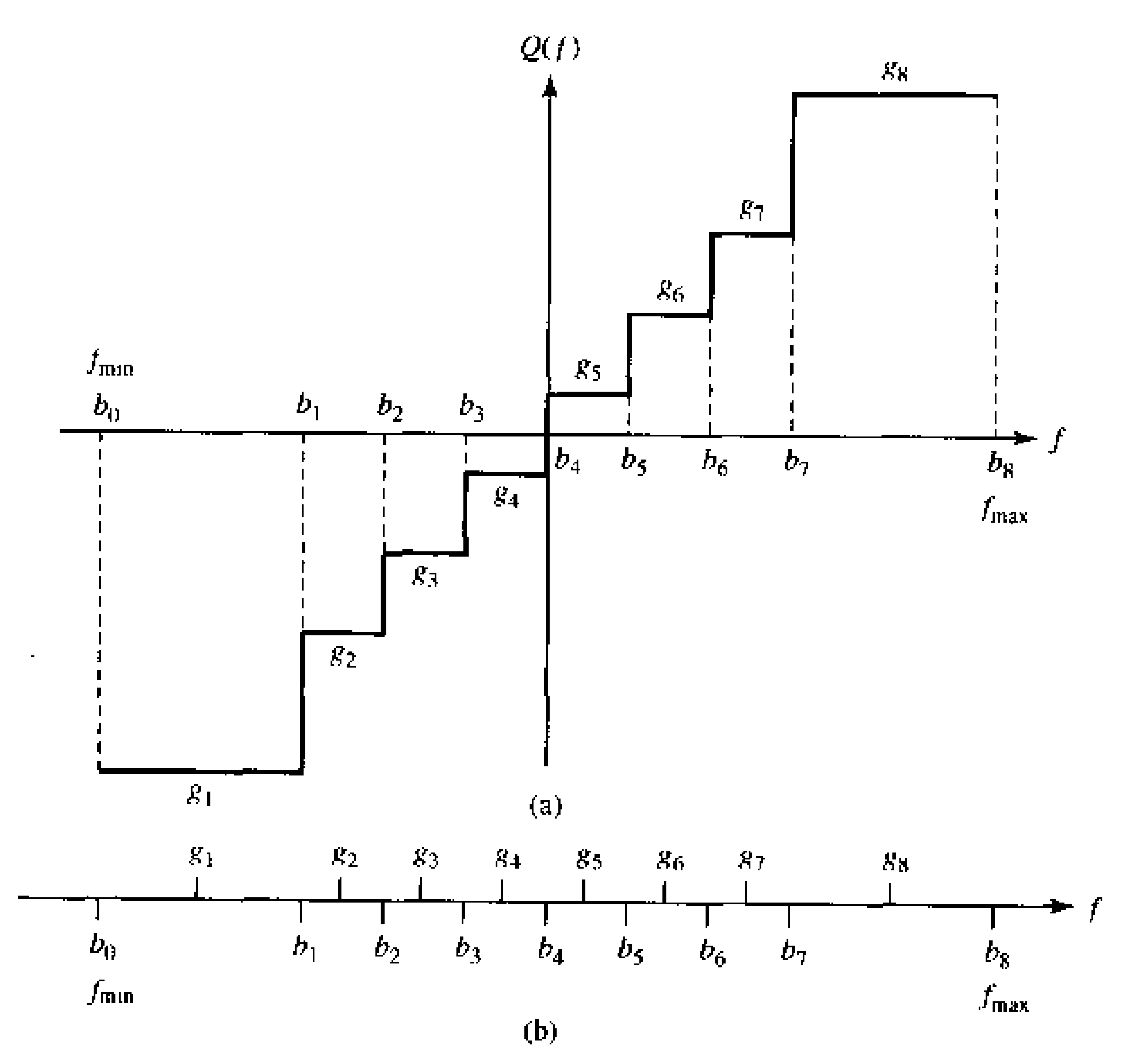


图 8.7　标量量化器的图示:(a)函数表示;(b)线分割表示

① 封闭的左界的使用是任意的,也可以使用$\mathcal{B}_l=(b_{l-1},b_l]$。

量化器的失真 在8.3.2节中,我们定义了当 N 个样点一起量化时原始信源与量化信源之间的失真测度。在标量量化器情况下,$N=1$,公式(8.3.12)所给出的失真测度变成:

$$D_q = E\{d_1(\mathcal{F}, Q(\mathcal{F}))\} = \int_{f\in\mathcal{B}} d_1(f, Q(f))p(f)\,df \tag{8.5.2}$$

$$= \sum_{l\in\mathcal{L}} P(\mathcal{B}_l) D_{q,l} \tag{8.5.3}$$

其中

$$D_{q,l} = \int_{f\in\mathcal{B}_l} d_1(f, g_l)p(f \mid f\in\mathcal{B}_l)\,df \tag{8.5.4}$$

在上述公式中,$\mathcal{F}$ 表示对应于信源$\mathcal{F}$中的任何样点的随机变量;$p(f), f\in\mathcal{B}$是$\mathcal{F}$的概率密度函数;$P(\mathcal{B}_l)=\int_{\mathcal{B}_l} p(f)\,df$ 是$f\in\mathcal{B}_l$ 的概率;而 $p(f|f\in\mathcal{B}_l)=p(f)/P(\mathcal{B}_l)$是给定$\mathcal{F}\in\mathcal{B}_l$ 时$\mathcal{F}$的条件概率密度函数。很明显,$D_{q,l}$表示在区域$\mathcal{B}_l$ 中所发生的平均失真。

当 $d_1(f,g)=(f-g)^2$,上述失真简化为均方误差,这等于量化误差$\mathcal{F}-Q(\mathcal{F})$的方差,由 σ_q^2 表示。具体地说,

$$\sigma_q^2 = E\{|\mathcal{F}-Q(\mathcal{F})|^2\} = \sum_{l\in\mathcal{L}} P(\mathcal{B}_l)\int_{b_{l-1}}^{b_l} (f-g_l)^2 p(f \mid \mathcal{B}_l)\,df \tag{8.5.5}$$

8.5.2 均匀量化

最简单的量化器是均匀量化器,它在相邻分界值之间和相邻重建值之间具有相等的距离:

$$b_l - b_{l-1} = g_l - g_{l-1} = q \tag{8.5.6}$$

其中,q 称为量化步长。这种量化器只适用于有限动态范围 B 的信号。令 f_{min}和 f_{max}表示最小和最大值,则 $B=f_{max}-f_{min}$。量化器是由量化电平数 L 或比特率 R 或量化步长 q 确定的。这些参数由 $q=B/L=B2^{-R}$互相联系。分界值和重建值由下式给出:

$$b_l = l * q + f_{min}, \quad g_l = (l-1)*q + q/2 + f_{min} \tag{8.5.7}$$

量化函数可以用封闭形式描述:

$$Q(f) = \left\lfloor \frac{f-f_{min}}{q} \right\rfloor * q + \frac{q}{2} + f_{min} \tag{8.5.8}$$

其中,$\lfloor x \rfloor$表示 x 的下限整数,也就是等于或小于 x 的最大整数(见图8.8)。

如果信源是均匀分布的,也就是说,如果:

$$p(f) = \begin{cases} 1/B & f\in(f_{min}, f_{max}) \\ 0 & \text{其他} \end{cases} \tag{8.5.9}$$

那么把公式(8.5.7)和(8.5.9)代入公式(8.5.5)得:

$$\sigma_q^2 = \frac{q^2}{12} = \sigma_f^2 2^{-2R} \tag{8.5.10}$$

其中 $\sigma_f^2=B^2/12$ 是原始信号$\mathcal{F}$的方差。量化器的SNR是:

$$\text{SNR} = 10\log_{10}\frac{\sigma_f^2}{\sigma_q^2} = (20\log_{10}2)R = 6.02R(\text{dB}) \tag{8.5.11}$$

因此,在均匀量化器中每增加一个比特就会对均匀信源的SNR产生6.02 dB的增益。这是量

化理论的一个著名结果。

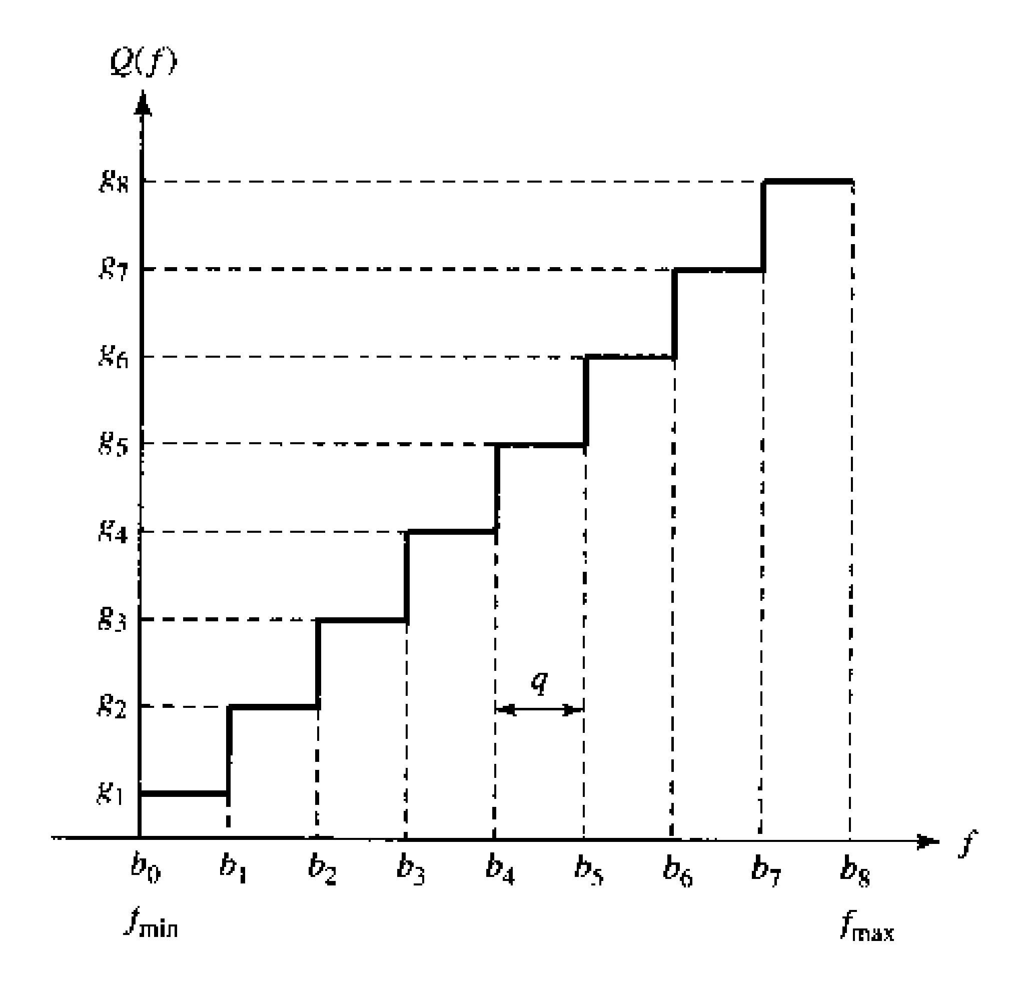


图 8.8　均匀量化器

8.5.3　最佳标量量化器

到目前为止所介绍的均匀量化器是易于实现的，但对于表示信源可能不是最有效的。在本节中，我们考察如何设计最佳标量量化器，以便在固定比特率、并且信源概率密度函数为 $p(f)$ 的条件下使失真达到最小。

最小均方误差（MMSE）量化器　在 MMSE 量化器中，选择 b_l 和 g_l 使得对于给定的 L，均方误差 σ_q^2 最小化。根据变分法，必要条件是 $\partial\sigma_q^2/\partial b_l=0$ 和 $\partial\sigma_q^2/\partial g_l=0, \forall\, l$。由公式(8.5.5)：

$$\frac{\partial\sigma_q^2}{\partial b_l}=\frac{\partial}{\partial b_l}\left(\int_{b_{l-1}}^{b_l}(f-g_l)^2p(f)df+\int_{b_l}^{b_{l+1}}(f-g_{l+1})^2p(f)df\right)$$

$$=(b_l-g_l)^2p(b_l)-(b_l-g_{l+1})^2p(b_l)=0$$

$$\frac{\partial\sigma_q^2}{\partial g_l}=\frac{\partial}{\partial g_l}\int_{b_{l-1}}^{b_l}(f-g_l)^2p(f)df=-\int_{b_{l-1}}^{b_l}2(f-g_l)p(f)df=0$$

简化上述公式得：

$$b_l=\frac{g_l+g_{l+1}}{2} \tag{8.5.12}$$

$$g_l=E\{\mathcal{F}\mid\mathcal{F}\in\mathcal{B}_l\}=\int_{\mathcal{B}_l}fp(f\mid f\in\mathcal{B}_l)df \tag{8.5.13}$$

条件平均 $E\{\mathcal{F}\mid\mathcal{F}\in\mathcal{B}_l\}$ 称为 $\mathcal{B}_l$ 的质心。

公式(8.5.12 及 8.5.13)的结果说明，最佳分界值处于两个最佳重建值的中点，而最佳重建值处于分界值之间区域的质心。注意，要求 $b_l=(g_l+g_{l+1})/2$ 等价于把任何 $f\in(g_l,g_{l+1})$ 量化成 g_l（如果 f 距离 g_l 比 g_{l+1} 更近）。更一般的叙述是，任何值 f 被量化为最接近 f 的重建值。

因此,公式(8.5.12)可等价地写成:

$$\mathcal{B}_l = \{f: d_1(f, g_l) \leqslant d_1(f, g'_l), \forall l' \neq l\} \tag{8.5.14}$$

公式(8.5.12)或公式(8.5.14)所给出的条件称为最邻近条件。另一方面,公式(8.5.13)称为质心条件。

可以证明若不用均方误差(MSE)测度而用失真测度,公式(8.5.14)所给出的最邻近条件仍然成立。然而,公式(8.5.13)中的质心条件必须改成:

$$g_l = \operatorname{argmin}_g \{E\{d_1(\mathcal{F}, g) \mid \mathcal{F} \in \mathcal{B}_l\}\} \tag{8.5.15}$$

其中,$\operatorname{argmin}_g\{E(g)\}$表示使函数 $E(g)$ 最小化的参量 g。公式(8.5.15)中所定义的 g_l 称为$\mathcal{B}_l$ 的广义质心,它的解取决于 $d_1(f,g)$的定义。例如,若 $d_1(f,g) = |f-g|$,则 g_l 是$\mathcal{B}_l$ 的中间值(见习题 8.12)。

MMSE 量化器的一个重要性质是它使不同分割区域中的量化误差相等,也就是说:

$$P(\mathcal{B}_l) D_{q,l} = D_q, \quad \forall l \in \mathcal{L} \tag{8.5.16}$$

最佳量化器的其他重要统计性质见参考文献[5,4]和习题 8.8。

均匀信源的 MMSE 量化器 把公式(8.5.9)中的均匀分布代入公式(8.5.12 及 8.5.13)将得到公式(8.5.7)中所给出的解。因此,如所期望的,均匀信源的 MMSE 量化器是均匀的。

非均匀信源的 MMSE 量化器 对于一个任意的概率密度函数 $p(f)$,公式(8.5.12 及 8.5.13)中的条件不总具有封闭形式的解。必须用数值程序来确定 b_l 和 g_l 的最佳集。参考文献[5]中的表 4.3 给出了具有单位方差和零均值的均匀、高斯、拉普拉斯和伽马信源的最佳 $\tilde{b}_l$ 和 $\tilde{g}_l$。可以证明(见习题 8.10)对于一个具有任意均值 μ_f 和方差 σ_f^2 的信源,最佳的 b_l 和 g_l 可由下式获得:

$$b_l = \sigma_f \tilde{b}_l + \mu_f, \quad g_l = \sigma_f \tilde{g}_l + \mu_f \tag{8.5.17}$$

非均匀信源的 MMSE 量化器一般情况下是非均匀,也就是说,分割区域在长度上是不等的。在这种情况下,量化值不能像均匀量化器那样用一个封闭形式的公式确定。一般地,必须把给定的值 f 与所有分界相比较,直到找到 $b_{l-1} \leqslant f < b_l$ 区域。或者说,可以寻找最接近 f 的 g_l。连续搜索需要高达 L 次的比较。用二叉树搜索程序,比较的次数可以减少到$\lceil \log_2 L \rceil$。

MMSE 量化器的渐进性能 对于具有任意概率密度函数的信源,用最邻近条件和质心条件,要导出量化器参数 b_l, g_l 以及相应的量化误差 σ_q^2 的封闭形式的解是困难的。然而,当比特率 R 和随之而来的量化级 $L=2^R$ 非常高时,可导出用概率密度函数表示的 MMSE 量化器封闭形式的解。有关的量化误差可以表示为:

$$\sigma_q^2 = \epsilon^2 \sigma_f^2 2^{-2R} \tag{8.5.18}$$

其中

$$\epsilon^2 = \frac{1}{12}\left(\int_{-\infty}^{\infty} \tilde{p}(f)^{1/3} df\right)^3 \tag{8.5.19}$$

其中 $\tilde{p}(f) = \sigma_f p(\sigma_f f)$表示一个具有单位方差的归一化信源的概率密度函数。公式(8.5.18)称为标量量化器性能的高率近似。把上述公式与公式(8.5.10)比较可以看出,对于均匀信源 $\epsilon^2 = 1$。对于非均匀信源一般$\epsilon^2 > 1$。特别地,对于高斯信源,$\epsilon^2 = 2.71$。回想 i.i.d. 高斯信源的失真率界限是由公式(8.3.20)给出的。因此,最好的标量量化器仍与该界限相差$\frac{1}{2}\log_2 \epsilon^2 =$

0.7191 比特/样点。对于其他几种信源的ϵ^2 值,见[5,表4.8]。为了接近于该界限,必须多个样点一起量化。对量化索引用可变长编码可减小因子ϵ^2;这将在 8.6.4 节中进一步讨论。

基于训练数据设计最佳标量量化器的劳埃德(Lloyd)算法　当欲量化信号的分布未知时,量化器可以基于含有欲量化的代表性样点的训练集进行设计。基于训练数据设计量化器的一个流行的方法是劳埃德算法[7,4]。如图 8.9 所示,该算法迭代地更新重建值和分界值。在每次迭代中,首先基于质心条件重新计算重建值,然后基于最邻近条件分割所有训练样点。在广义质心和失真计算中所需要的统计期望用样点的平均近似。在 MSE 准则下,一个分割的重建值简单地取为属于该分割区域的训练样点的平均值,而分界值简单地取为两个相邻重建值的平均。在图中,K_l 表示 B_l 中的样点数。

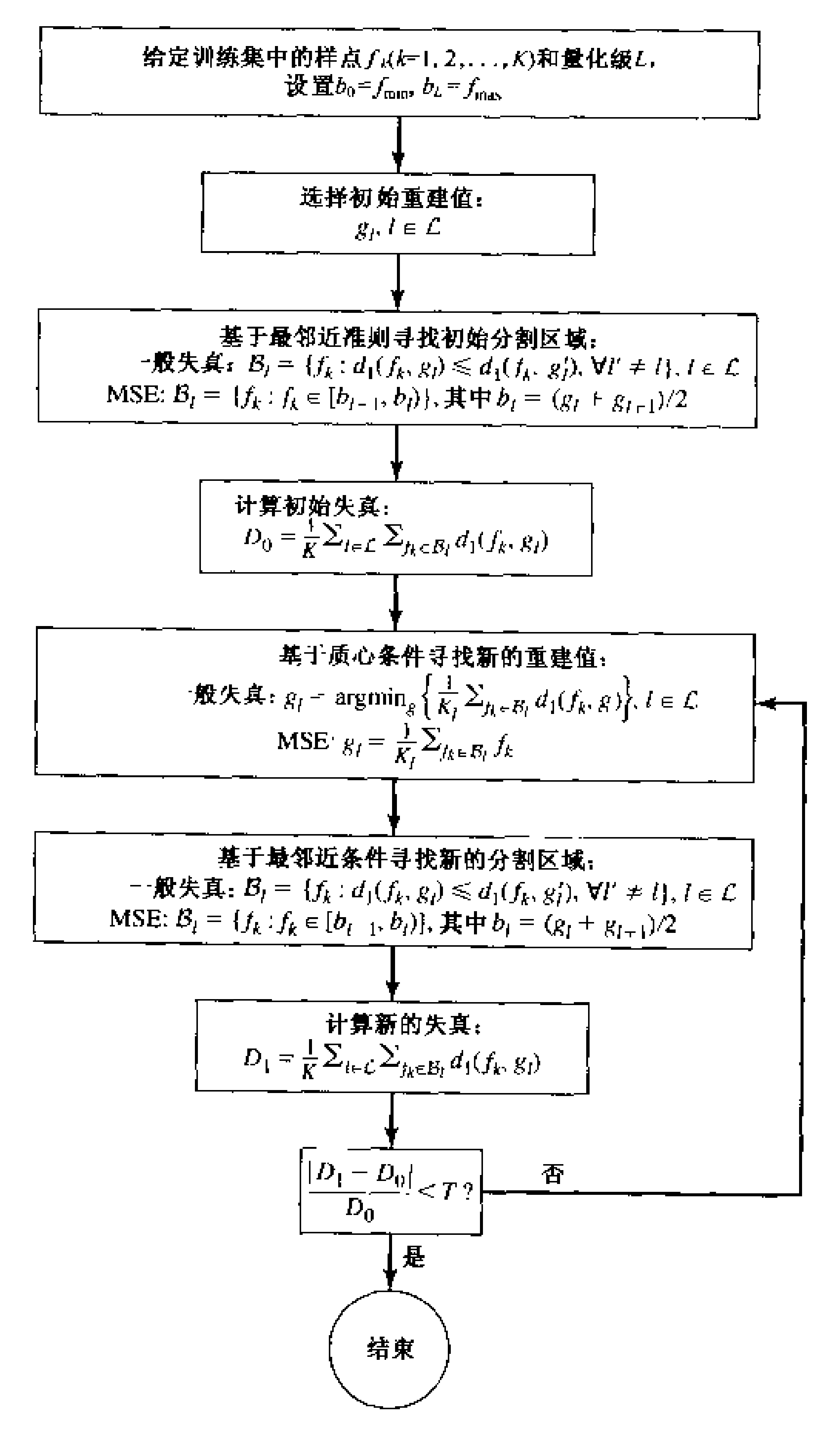


图 8.9　用训练数据设计最佳标量量化器的劳埃德算法

8.6 矢量量化

8.6.1 基本原理

我们可以把一次量化一个样点换成一次量化一组 N 个样点。每个样点组称为一个矢量，而这个一次量化一个矢量的过程称为矢量量化(VQ)。在图像或视频情况下，一个矢量通常对应一个像素块。采用 VQ 的动机主要源于这种观察，即在典型的图像或视频帧中，块中的样点是相互联系的，使得一些方块模式比其他的更为相似。VQ 的任务从本质上说是要找到 L 个最通用的模式，并且用其中一个与原始模式最相似的代表模式来代替任何给定的块。

上述讨论假设一个矢量对应于一组样点值。在每个样点由多个值描述的信源中，每个样点本身就可以用一个矢量量化器进行量化。例如，彩色图像中的每个样点包含三个分量。为了在一个采用 8 比特图形卡的计算机监视器上显示 24 比特彩色图像，我们必须找到 256 种彩色使它们能最好地表示所有可能的 2^{24} = 1 600 万种彩色。

把每个可能的 N 维矢量 $\mathbf{f}$ 看做 N 维空间的一点；VQ 问题就是要把 N 维空间分割成 L 个区域 $\mathcal{B}_l$，并由具有代表性的点 $\mathbf{g}_l$ 表示区域 $\mathcal{B}_l$ 中所有的点。图 8.10 图示了 $N=2$ 情况下的这一过程。我们称 $\mathcal{B}_l$ 为分割区域，$\mathbf{g}_l$ 为重建矢量或码字。包含所有码字 $C=\{\mathbf{g}_l, l\in\mathcal{L}\}$ 的集称为码书[①]。量化函数可以描述为：

$$Q(\mathbf{f}) = \mathbf{g}_l, \qquad \mathbf{f}\in\mathcal{B}_l \tag{8.6.1}$$

如果用固定长度的二进制编码把重建码字转成二进制比特，那么具有 N 个样点的每一组需要 $\lceil \log_2 L \rceil$ 个比特，且比特率(比特/样点)是：

$$R = \frac{1}{N}\lceil \log_2 L \rceil \tag{8.6.2}$$

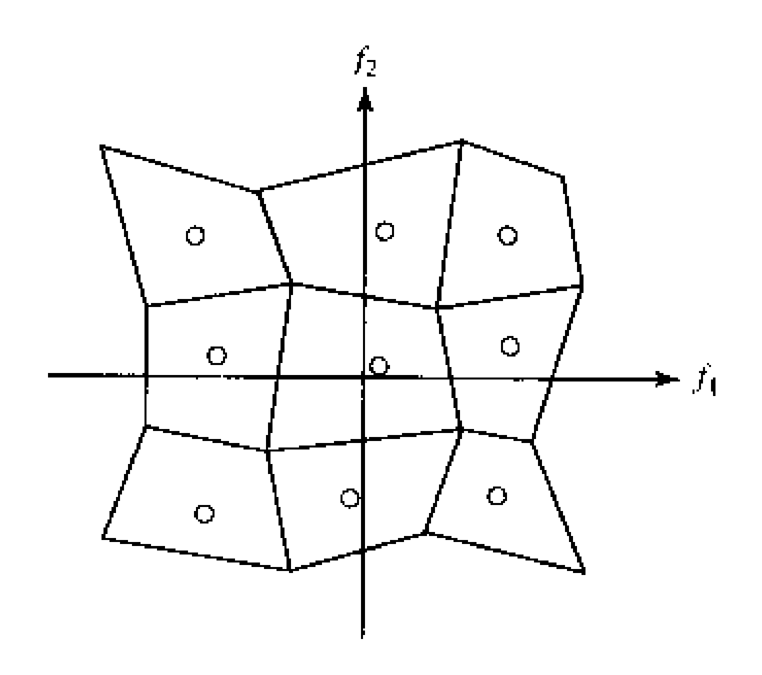


图 8.10　矢量维 $N=2$ 的矢量量化图示。每个分割区域内的空心圆圈表示该区域的重建码字

① 注意，为了用二进制比特表示量化信号，必须把每个重建码字映射为二进制码字。在本节中的“码字”和“码书”是指重建码字和码书，而 8.3.1 节和 8.4 节中它们是指二进制码字和码书。

比较公式(8.6.1)与公式(8.5.1)以及图 8.10 与图 8.7(b),可以看出标量量化器只是矢量量化器的一种特殊情况,在那里,$\mathbf{f}$ 是标量,而$\mathcal{B}_l$ 是线段。一般地,与标量情况一样,分割区域$\mathcal{B}_l$ 不能简单地由少量判定值描述。

为了计算矢量量化器引入的量化误差,令$\mathcal{F}$ 和 $Q(\mathcal{F})$表示原始矢量和量化后的矢量,且$p_N(\mathbf{f})$表示$\mathcal{F}$ 中各分量的联合概率密度函数。应用 8.3.2 节中的表示法,公式(8.6.1)所描述的矢量量化器的条件概率 $q_N(\mathbf{g}|\mathbf{f})$是:

$$q_N(\mathbf{g} \mid \mathbf{f}) = \begin{cases} 1 & \mathbf{f} \in \mathcal{B}_l, \mathbf{g} = \mathbf{g}_l \\ 0, & \text{其他} \end{cases} \tag{8.6.3}$$

在这种情况下,公式(8.3.12)中的失真准则简化为:

$$D_q = E\{d_N(\mathcal{F}, Q(\mathcal{F}))\} = \int_{\mathcal{B}} p_N(\mathbf{f}) d_N(\mathbf{f}, Q(\mathbf{f})) d\mathbf{f} \tag{8.6.4}$$

$$= \sum_{l=1}^{L} P(\mathcal{B}_l) D_{q,l} \tag{8.6.5}$$

其中

$$D_{q,l} = E\{d_N(\mathcal{F}, Q(\mathcal{F})) \mid \mathcal{F} \in \mathcal{B}_l\} = \int_{\mathbf{f} \in \mathcal{B}_l} p_N(\mathbf{f} \mid \mathbf{f} \in \mathcal{B}_l) d_N(\mathbf{f}, \mathbf{g}_l) d\mathbf{f} \tag{8.6.6}$$

最邻近量化器和 VQ 的复杂度　一般地,一个矢量量化器由码字 $\mathbf{g}_l$ 和分割区域$\mathcal{B}_l$ 确定。用最邻近量化器,对于任何输入矢量 $\mathbf{f}$,通过将 $\mathbf{f}$ 与所有码字进行比较,并且在距离测度 $d_N(\mathbf{f},\mathbf{g}_l)$ 条件下寻找与它最接近的一个来确定量化矢量。换句话说,分割区域通过下式由码字隐含地确定:

$$\mathcal{B}_l = \{\mathbf{f} \in \mathcal{R}^N : d_N(\mathbf{f}, \mathbf{g}_l) \leqslant d_N(\mathbf{f}, \mathbf{g}'_l), \forall l' \neq l\} \tag{8.6.7}$$

正如将在 8.6.3 节中说明的,这是最小化平均量化误差的必要条件。本质上,实际中所用的所有量化器都属于这一类;图 8.11 说明了最邻近量化器的操作。

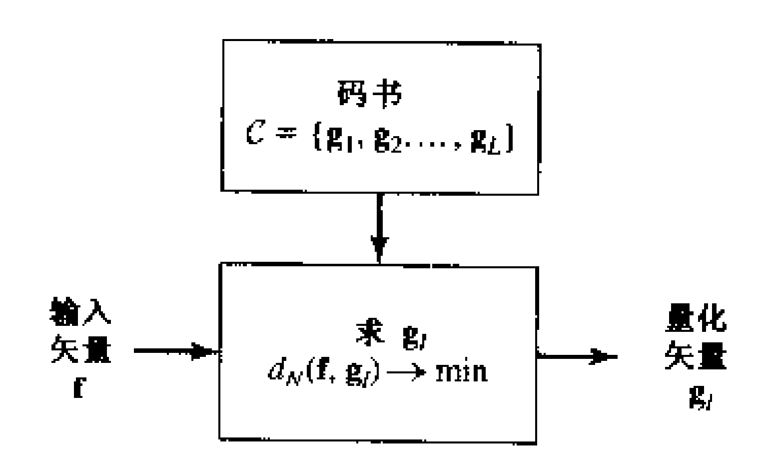


图 8.11　最邻近量化器的操作

用最邻近量化器,对于每一个输入矢量,必须计算这个矢量与所有 L 个重建码字之间的失真。用平方误差测度:

$$d_N(\mathbf{f}, \mathbf{g}) = \frac{1}{N} \sum_{n=1}^{N} (f_n - g_n)^2 \tag{8.6.8}$$

每次计算都需要 N 次运算(一个运算包括一次减法、一次乘法和一次加法)。因此,量化一个矢量所需要的总运算次数为 NL。从公式(8.6.2)可知,$L = 2^{NR}$。因此,运算次数是 $N2^{NR}$。就存储需求而言,每个码字需要 N 个基本存储单元,而所有 L 个码字需要 $NL = N2^{NR}$个存储单

元。因此,VQ 的复杂度和存储需求随矢量维数呈指数增长。回想定理 8.4 告诉我们,一般只有当 N 趋于无穷时才可达到信源的 RD 界限。一般地说,N 越大,编码器的效率就越高。然而,复杂度随 N 呈指数增长限制了实际可以使用的矢量维数。考虑视频帧的量化,每 N 个像素的块用长度为 L 的码书量化。假设帧率是 f_t 且帧的大小为 $N_x \times N_y$,那么每秒的运算次数将是 $(f_t N_x N_y / N) \cdot N2^{NR} = f_t N_x N_y 2^{NR}$。对于 BT.601 视频的 Y 分量,$f_t = 30$, $N_x = 720$, $N_y = 480$,而 4×4大小的矢量和 1 比特/样点的比特率需要每秒 $6.8E+11$ 的运算次数!对于图像和视频编码,N 通常限制于 4×4 或以下。

为了降低复杂度,类似于 SQ 情况的二叉树搜索,如果每个分割区域是一个多面体,则不需要进行穷尽搜索。在这种情况下,不同的分割区域是用横断的超平面定义的。可以将给定的矢量与一个超平面进行比较,每次去掉平面一个边上的码字。这样,可以把比较的次数缩减到远小于 L,但可能不会低到 $\log_2 L$,除非码书具有特殊的树结构。为了进一步降低复杂度,已开发了各种快速 VQ 方案,这些方案对码书结构加上了一定的约束,从而减少了运算次数。Gersho 和 Gray[4]提供了各种 VQ 技术的广泛内容。

8.6.2 点阵矢量量化器

如在 8.5.2 节所学习过的,最简单的标量量化器是均匀量化器;其中,一定范围的实线被分割成相等长度的量化箱(bin)。在 VQ 情况下类似的均匀量化器是点阵量化器;其中,所有分割区域都具有相同的形状和大小,而且实际上都是基本区域的移位形式,基本区域是点阵的沃罗纳(Voronoi)区域。图 8.12 示出了两种点阵量化器,分别采用矩形和六边形点阵。在第 3 章中,我们介绍了在多维空间上均匀采样的点阵概念。这里我们看到也可以用点阵来定义多维空间上的均匀量化器。

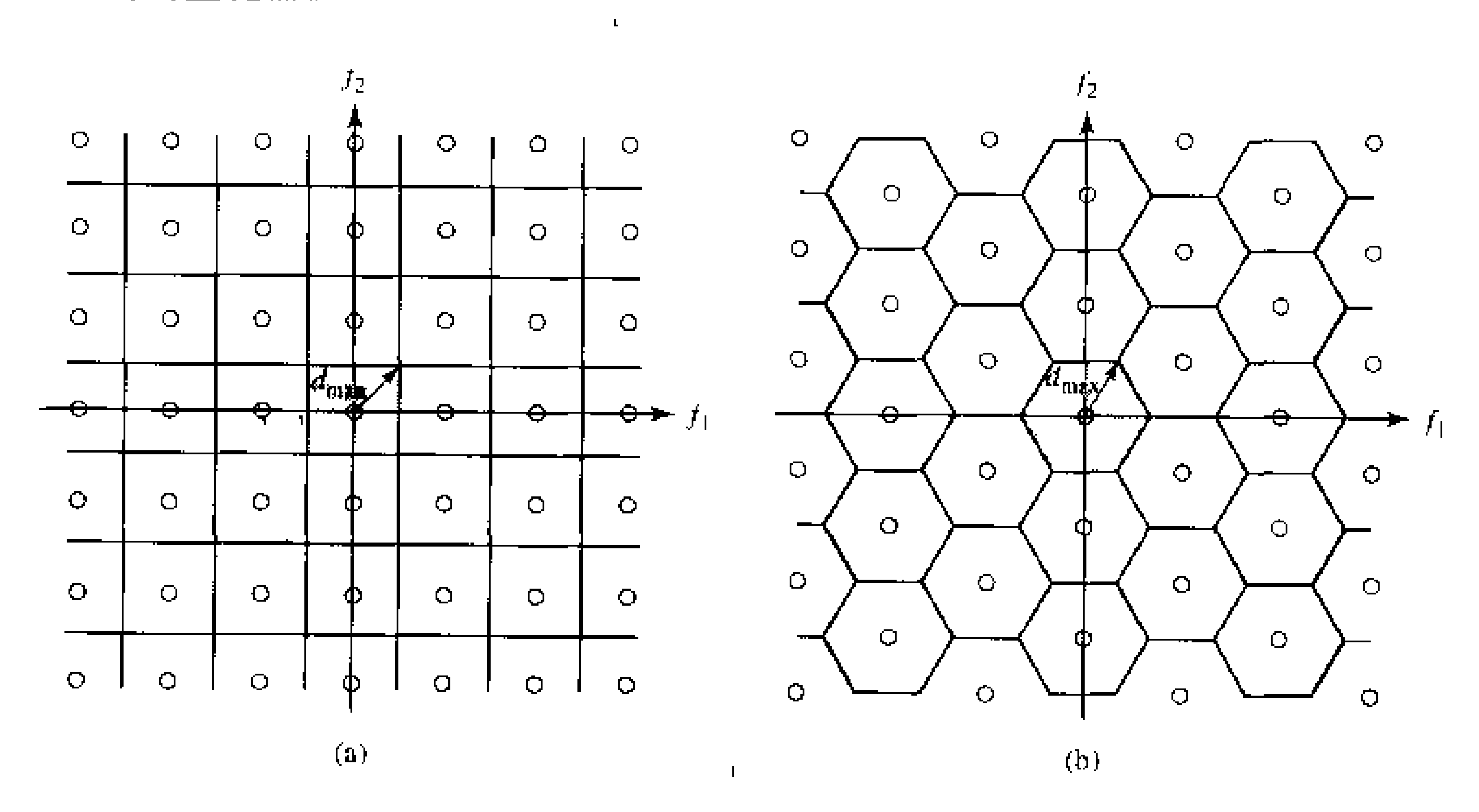


图 8.12 使用(a)矩形和(b)六边形点阵的点阵量化器。阴影区域表示沃罗纳区域;$d_{\max}$是输入矢量与它的量化形式之间的最大距离

回想 N 维实空间$\mathcal{R}^N$ 上的点阵 Λ 是由一个基本矢量集 $\mathbf{v}_n$ ($n = 1,2,\cdots,N$),或一个生成矩

阵$[\mathbf{V}] = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_N]$定义的。点阵中的点是那些可以表示为下式的矢量：

$$\mathbf{g}_l = \sum_{n=1}^{N} m_{l,n} \mathbf{v}_n = [\mathbf{V}] \mathbf{m}_l \tag{8.6.9}$$

其中，$m_{l,n} \in \mathcal{Z}$是整数或$\mathbf{m}_l \in \mathcal{Z}^N$是一个 N 维整数矢量。通常，一个码字 $\mathbf{g}_l$ 由它所对应的 $\mathbf{m}_l$ 索引。Λ 的沃罗纳区域$\mathcal{V}$是其中所有的点更接近原点而不是任何其他非零阵点的区域。对于一个点阵量化器，码书由点阵中的所有点组成，或是由点阵的陪集中的所有点组成，点阵的陪集是点阵的一个移位形式。这样的量化器有无穷多的码字。实际上，更常用的是点阵或它的陪集的截断形式，它仅取$\mathcal{R}^N$上的某个支撑区内的样点。

对于一个$\mathcal{R}^N$中的某个有限区域上的均匀分布信源，如果我们假设所有的分割区域恰好填满支撑区[①]，那么所有分割区域可能是相等的，且每个区域的失真与沃罗纳区域上的失真相同。回想沃罗纳区域$\mathcal{V}$的容积等于$|\det[\mathbf{V}]|$。这样，条件概率密度函数 $p(\mathbf{f}|\mathbf{f} \in \mathcal{V}) = 1/|\det[\mathbf{V}]|$。因此，平均量化误差是：

$$D_q = D_{q,l} = \frac{1}{|\det[\mathbf{V}]|} \int_{\mathcal{V}} d_N(\mathbf{f}, \mathbf{0})\, d\mathbf{f}. \tag{8.6.10}$$

根据 MSE 准则，我们有：

$$D_q = \frac{1}{|\det[\mathbf{V}]|} \int_{\mathcal{V}} \|\mathbf{f}\|^2 d\mathbf{f}. \tag{8.6.11}$$

几种高维点阵的 D_q 值已经由 Conway 和 Sloane[2]制成了表。

值得注意的是，即使对于 i.i.d. 信源，VQ 也可以改善 SQ 的编码效率。这是因为 VQ 允许用分割区域填满空间时有更多的灵活性。考虑图 8.12 所给的两个例子。在每一种情况下，用区域的中心(码字)表示区域中的点的最大失真是 $d_{max} = 1$，从而两个量化器将产生相同的最大失真。但为了填满相同的二维空间，需要较少的六边形，因为六边形点阵的沃罗纳区域大于矩形点阵的沃罗纳区域。特别地，在这个例子中矩形和六边形点阵的沃罗纳区域的面积分别是 $2d_{max}^2$ 和 $2.59d_{max}^2$。在图中所考虑的整个支撑区中，矩形点阵具有 36 个以上的分割区域，而六边形点阵仅有 30 个区域[②]。目前，矩形点阵量化器也可以通过在每一维应用一个均匀标量量化器来实现。这意味着当两个样点相互独立且都是均匀分布时，用一个设计得很好的矢量量化器把它们作为矢量一起量化可以减少比特率。

点阵量化器的实现 如前面所述，对于一个任意的最邻近量化器，量化一个给定的矢量 $\mathbf{f}$ 包括一个穷尽搜索。用点阵量化器就没这个必要。在这种情况下，我们可以首先用下式确定一个量化矢量的实索引矢量：

$$\mathbf{m} = [\mathbf{V}]^{-1} \mathbf{f} \tag{8.6.12}$$

然后，可以计算与邻接 $\mathbf{m}$ 的所有整数索引矢量相关的失真，取 $\mathbf{m}$ 的每个分量的下限整数和上限整数，并确定哪个给出最小失真。为了进一步降低复杂度，也可以简单地把 $\mathbf{m}$ 四舍五入到最近的整数矢量。读者可参阅参考文献[2]以便获得点阵编码算法和性能的更详细的讨论。

---

① 一般情况下这是不成立的，除非使用超立方体点阵。但当 L 很大时，支撑区边界上的分割区域数很小，从而那些区域的影响可以忽略。

② 回想第 3 章我们知道六边形点阵在覆盖空间时更有效。

8.6.3 最佳矢量量化器

尽管点阵量化器易于实现，但当信源不是均匀分布时，它们可能不是最有效的。对于一个给定的 N 维矢量和信源概率密度函数 $p_N(\mathbf{f})$，希望设计一种最佳量化器，对于给定的比特率 R 或(等价地)码书大小 L，使给定的失真准则最小化。理想情况下，应该同时确定码字 $\mathbf{g}_l$ 和分割区域$\mathcal{B}_l$ 以使平均失真 D_q 最小。然而，最优化解难以找到。替换的方法是，可以分为两个阶段进行。首先，对于给定的码字 $\mathbf{g}_l(l\in\mathcal{L})$，分割区域$\mathcal{B}_l(l\in\mathcal{L})$应该使所有区域上的平均失真 D_q 为最小。其次，对于给定的分割区域$\mathcal{B}_l$，码字 $\mathbf{g}_l$ 的选择应该使$\mathcal{B}_l$ 内的失真 $D_{q,l}$为最小。

对于第一个问题，从公式(8.6.4)容易看出：如果 $Q(\mathbf{f})$的选择使 $d_N(\mathbf{f},Q(\mathbf{f}))$ 对于任何 $\mathbf{f}$ 都为最小，则 D_q 将被最小化。也就是说：

$$Q(\mathbf{f}) = \mathbf{g}_l = \operatorname{argmin}_{\mathbf{g}'_l, l'\in\mathcal{L}}\{d_N(\mathbf{f},\mathbf{g}'_l)\} \tag{8.6.13}$$

换句话说，$\mathbf{f}$ 应该量化到最接近它的码字。这种分配等价于按照下式建立分割区域$\mathcal{B}_l$：

$$\mathcal{B}_l = \{\mathbf{f}: d_N(\mathbf{f},\mathbf{g}_l) \leqslant d_N(\mathbf{f},\mathbf{g}'_l), \forall\, l' \neq l\} \tag{8.6.14}$$

我们称公式(8.6.13)或公式(8.6.14)为最邻近条件。

现在，我们考虑第二个问题。很明显，应该选择 $\mathbf{g}_l$ 使得公式(8.6.6)中的 $D_{q,l}$最小化，也就是说：

$$\mathbf{g}_l = \operatorname{argmin}_{\mathbf{g}} E\{d_N(\mathcal{F},\mathbf{g}) \mid \mathcal{F} \in \mathcal{B}_l\} \tag{8.6.15}$$

我们称这个 $\mathbf{g}_l$ 为$\mathcal{B}_l$ 的广义质心，并且公式(8.6.15)是质心条件。

当失真测度是 MSE 时，

$$D_{q,l}(\mathbf{g}) = \frac{1}{N}E\{\|\mathcal{F} - \mathbf{g}\|^2 \mid \mathcal{F} \in \mathcal{B}_l\} = \frac{1}{N}\int_{\mathcal{B}_l}\|\mathbf{f} - \mathbf{g}\|^2 p(\mathbf{f} \mid \mathbf{f}\in \mathcal{B}_l)d\mathbf{f} \tag{8.6.16}$$

使 $D_{q,l}$最小的 $\mathbf{g}_l$ 必须满足$\partial D_{q,l}/\partial\mathbf{g}=0$。由此得到：

$$\mathbf{g}_l = \int_{\mathcal{B}_l} p(\mathbf{f} \mid \mathbf{f}\in \mathcal{B}_l)\mathbf{f}d\mathbf{f} = E\{\mathcal{F} \mid \mathcal{F} \in \mathcal{B}_l\}. \tag{8.6.17}$$

也就是说，$\mathbf{g}_l$ 是$\mathcal{B}_l$ 的条件平均或质心。公式(8.6.14 及 8.6.15)定义了一个对于任意失真测度的最优矢量量化器，而公式(8.6.14)和(8.6.17)一起定义了一个 MMSE 矢量量化器。

注意，最邻近条件和质心条件是最小化 D_q 的必要而非充分条件。满足两个条件的量化器仍可能没达到 D_q 的总体最小化。图 8.13 示出了两个用于均匀分布信号的二维矢量量化器。很容易证明每个量化器都满足这两个条件，然而很明显右边的量化器将具有较低的 MSE。因此，左边的量化器仅为局部最优。

设计最佳矢量量化器的广义劳埃德算法 对于一个任意的信源，难以导出 $\mathbf{g}_l,\mathcal{B}_l(l\in\mathcal{L})$ 的封闭形式的解。由广义劳埃德算法可以获得局部最优解[①]。与 SQ 的劳埃德算法一样，VQ 的广义劳埃德算法基于训练数据迭代地确定最佳的 $\mathbf{g}_l$ 和$\mathcal{B}_l$。图 8.14 给出这个算法的流程表。将这个图与图 8.9 比较，很容易看出这两个算法的相似性。

① 按照首先把劳埃德算法推广到矢量量化的作者而命名，这个算法也称为 LBG 算法[6]。同样的算法也已经应用于把数据点自动聚集成几类的模式识别中；在那里，称它为 K 平均算法。

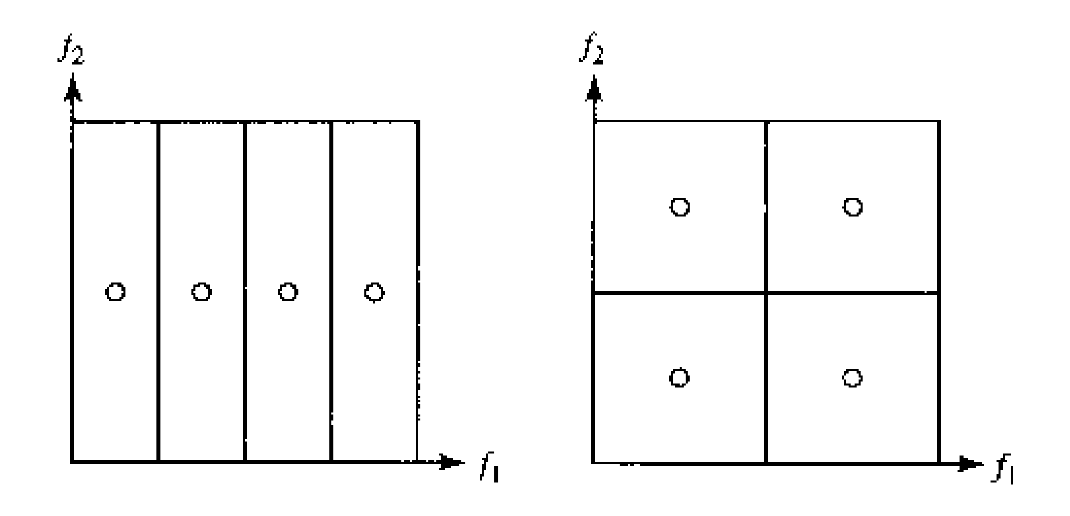


图 8.13　满足最邻近和质心条件的两个矢量量化器，而右边的量化器具有较低的失真

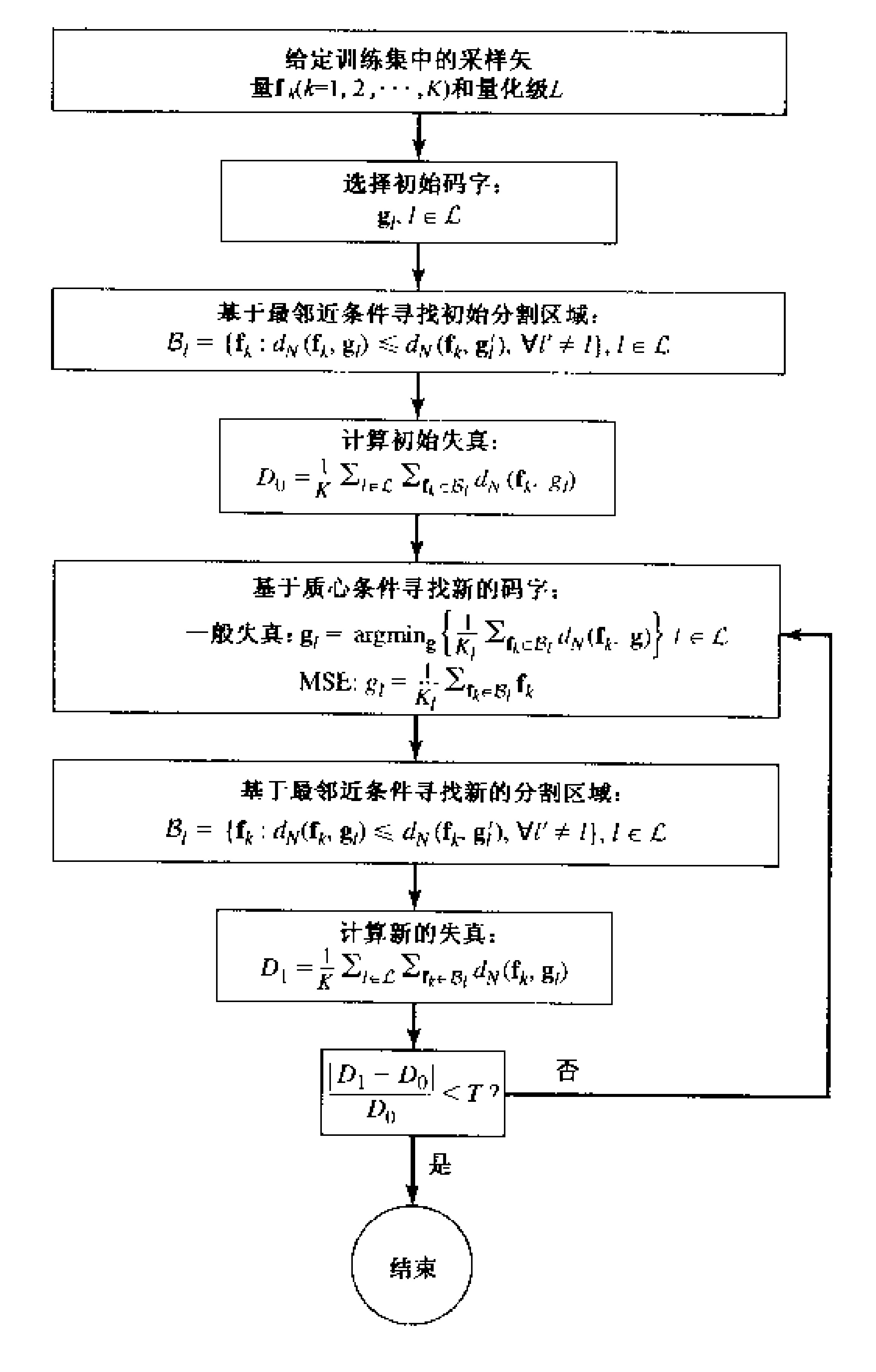


图 8.14　用训练数据设计最佳矢量量化器的广义劳埃德算法

例 8.5 （矢量量化器设计）

图 8.15 给出在 MSE 准则下使用广义劳埃德算法设计量化器的一个例子。在每次迭代中，我们首先通过对区域内的所有训练矢量取平均来对每个以前所确定的分割区域重新计算质心矢量(用圆表示)，然后根据训练矢量到更新的质心矢量的距离重新分割(用椭圆表示)所有的训练矢量。图中示出了前三次迭代的结果。因为第三次迭代后的分割仍与第二次迭代后的相同，所以第四次迭代将不会改变码字或分割区域。因此，第三次迭代后所给出的结果是最终解。

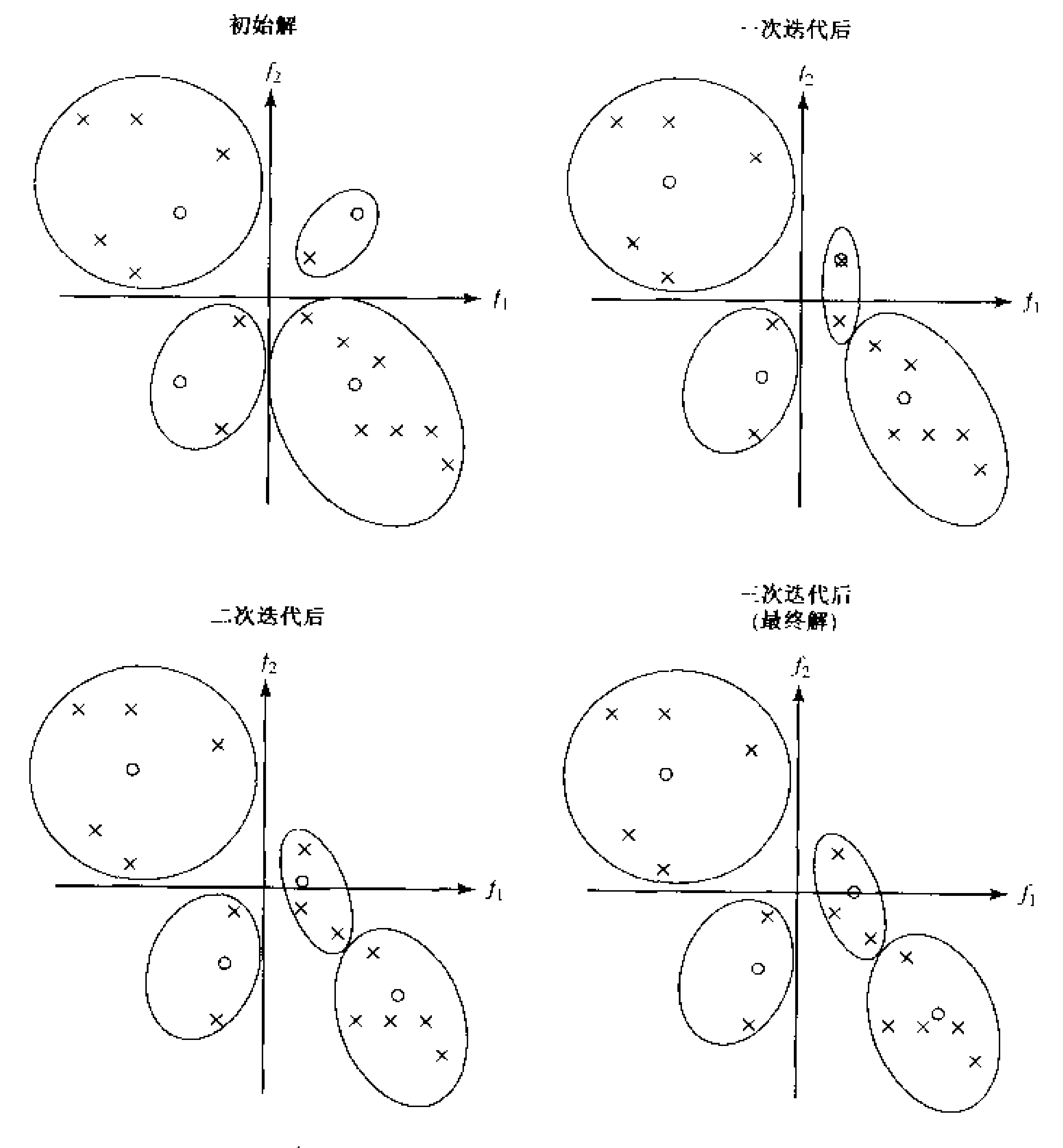


图 8.15 用广义劳埃德算法设计矢量量化器的一个例子。×表示训练矢量，O 表示码字(见例 8.5)

由广义劳埃德算法得到的量化器取决于初始码字。当初始码字选择得不合适时，算法可能就会收敛于局部最小而远离全局最小。选择初始码字的一个简单且相当有效的方法是采用在每维应用均匀标量量化所得到的码字。关于 VQ 设计算法广泛的讨论，包括初始码字的选择见参考文献[4]。

8.6.4　熵约束最佳量化器的设计

到目前为止，我们假设长为 L 的码书中的每个码字取$\lceil \log_2 L \rceil$个比特来表示。因为重建码字通常以不同的概率出现，所以可以用哈夫曼编码或别的可变长编码技术来进一步降低比特率。在 8.5.3 节和 8.6.3 节所描述的最佳量化器的设计方法中，我们试图通过假设 R 与 L 具有的公式(8.6.2)关系，使在给定比特率 R 或码书长度 L 时失真最小。如果用可变长编码技术对量化后的矢量进行编码，就 $\mathbf{g}_l$ 和$\mathcal{B}_l$ 而言所得的解不再是最佳的。采用熵约束最佳量化器设计，我们以码率约束为条件使失真最小化，这里假设码率等于量化信源的熵。如果我们一次编码一个码字，那么比特率由下式约束：

$$\bar{R}_N = -\frac{1}{N}\sum_{l\in\mathcal{L}} P(\mathcal{B}_l)\log_2 P(\mathcal{B}_l) \tag{8.6.18}$$

因为码字 $\mathbf{g}_l$ 的概率是 $P(\mathcal{B}_l)$。给定期望的比特率 R 和矢量的维数 N，最优化问题就是在下式的约束下使 D_q 为最小：

$$-\sum_{l\in\mathcal{L}} P(\mathcal{B}_l)\log_2 P(\mathcal{B}_l) \leqslant RN \tag{8.6.19}$$

用拉格朗日乘子法，这个有约束的最优化问题可以转化为无约束的最优化问题：

$$\text{minimize} J(\lambda) = D_q + \lambda\Big(-\sum_{l\in\mathcal{L}} P(\mathcal{B}_l)\log_2 P(\mathcal{B}_l) - NR\Big) \tag{8.6.20}$$

其中，λ 的选择必须满足公式(8.6.19)中的约束。对于每个固定的 λ，用广义劳埃德算法，将失真准则从 D_q 改变为公式(8.6.20)，就可以得到 $\mathbf{g}_l$ 和$\mathcal{B}_l$ 的解。然后可以用公式(8.6.18)计算这个解所需要的比特率。实际上，不同 λ 值得到的解对于不同比特率是最优的。

在熵约束下，当 L 很大时，非均匀信源的最佳量化器趋于均匀。这样，每个分割区域中的最大失真是相似的。然而，因为对应于较高概率密度函数区域的码字比其他码字出现的可能性更大，所以将用较短的码字编码。

注意，上述讨论同样适用于标量量化。回想任何非熵约束的 MMSE 量化器的率失真函数可以在高码率情况下写成公式(8.5.18)的形式。应用熵编码的净效果是减小因子ϵ^2。对于高斯信源，用熵约束 MMSE 量化器和标量无损编码，ϵ^2 可以从 2.71 降到 1.42，这对于相同的失真相当于节省 0.467 比特/样点。与 i.i.d. 高斯信源的 RD 界限比较，这表示与该界限仅差 0.255 比特/样点[5,4.6.2 节]。

8.7　小结

编码系统的一般结构(8.1 节)

- 编码系统通常由三部分组成(图 8.1)：基于信源模型的分析，把信号样点转换成模型参数；参数的量化(对于有损编码系统)；以及量化参数的二进制编码。
- 不同编码系统之间的主要区别在于所用的信源模型。视频编码方法可以分为两类：一类采用直接表征像素值(基于波形的)的信源模型，一类采用描述所包含的物体结构(基于内容的)的模型。

无损和有损编码的界限(8.3节)

- 离散信源的无损编码所需要的最小码率是由信源熵率确定的(定理8.2)。
- 在给定失真(RD界限)下,编码所需要的最低码率是由原始和量化信源之间的最小互信息确定的(定理8.4)。
- 只有无限多个样点一起编码才能达到上述界限。
- 在具有相同方差的所有信源中,高斯信源需要最高的比特率。

二进制编码(8.4节)

- 二进制编码可以一次应用于一个样点或一组样点,具有或不具有基于前面样点的条件。
- 哈夫曼编码和算术编码是可变长编码的两种最流行的方法。哈夫曼编码器的设计和操作较简单,但不能达到具有合理复杂度的无损编码的界限。也难于使哈夫曼编码器适应信号统计特性的变化。算术编码器能够更容易达到熵界限,且对非平稳信号更有效,但它们的实现也更复杂。

标量量化(8.5节)

- MSE最优化的必要条件是质心条件(公式(8.5.13)或(8.5.15))和最邻近条件(公式(8.5.12)或(8.5.14))。局部最优量化器可以用劳埃德算法进行设计(图8.9)。
- 对于MMSE量化器,当码率足够高时,信源的运算RD函数具有公式(8.5.18)的一般形式。因子ϵ^2取决于信源的概率密度函数。
- MMSE量化器假设用固定长编码对量化索引进行编码。用可变长编码并且当码率足够高时,均匀量化接近最佳。对于图像和视频编码,均匀量化后跟VLC比非均匀量化更经常被采用。

矢量量化(8.6节)

- VQ可以利用样点之间的相关性,并且它的效率随矢量维数的增加而增加。当矢量维数趋于无穷时,有可能达到RD界限。
- 一般的无条件的VQ编码器的计算复杂度和存储需求随矢量维数的增加而呈指数增加。实际的图像和视频编码器用小于等于4×4的矢量尺寸。
- MSE最优化的必要条件是质心条件(公式(8.6.17)或(8.6.15))和最邻近条件(公式(8.6.14))的广义形式。局部最佳矢量量化器可以用广义劳埃德算法进行设计(图8.14)。

8.8 习题

8.1 考虑一个具有字母表$\mathcal{A}=\{a_1,a_2,\cdots,a_L\}$的离散信源。计算以下两种情况下信源的熵:

(a) 信源是均匀分布的,具有$p(a_l)=1/L,\ \forall\, l\in\mathcal{L}$。

(b) 对于一个特殊的$k\in\mathcal{L}$,$p(a_k)=1$和$p(a_l)=0,\ l\neq k$。

8.2 证明公式(8.2.6~8.2.12)所给出的关系。

8.3　证明公式(8.2.17～8.2.19)所给出的关系。

8.4　证明均值为 η，方差为 σ^2 的一个高斯 RV 的微分熵如公式(8.2.23)所示。

8.5　说明 N 阶矢量编码比($N-1$)阶条件编码差，如果：

$$\frac{1}{N}H_N(\mathcal{F}) - H_{c,N-1}(\mathcal{F}) > \frac{N-1}{N}$$

8.6　一个有三个符号 $\mathcal{A}=\{a_1,a_2,a_3\}$ 的马尔可夫信源具有如下的概率分布：

$$p(a_i) = \frac{1}{3};\quad i = 1,2,3$$

$$P\{a_i/a_j\} = \begin{cases} \frac{1}{2} & i = j \\ \frac{1}{4} & \text{其他} \end{cases}$$

(a) 计算此信源的一阶熵、二阶熵、一阶条件熵和熵率。

(b) 为此信源设计一阶、二阶和一阶条件哈夫曼码。计算每种情况产生的比特率。

(c) 此信源可以达到的最小比特率是多少？如何达到这个最小比特率？

8.7　用算术编码来编码和解码以下序列。用序列中每种符号出现的频率作为该符号概率的估计。

信源序列：a c b a a b a c a c b a

编码序列的比特率是多少？把此结果与标量哈夫曼编码相比较。

8.8　说明对于 MMSE 量化器，原始的随机变量 $\mathcal{F}$、量化后的随机变量 $\mathcal{G}$ 以及量化误差 $\mathcal{Q}=\mathcal{F}-\mathcal{G}$ 满足如下的统计关系：

(a) 量化值是原始值的无偏估计：$E\{\mathcal{G}\}=E\{\mathcal{F}\}$。

(b) 量化值与量化误差正交：$E\{\mathcal{G}\ \mathcal{Q}\}=0$。

(c) 量化过程减小了信号方差：$\sigma_{\mathcal{G}}^2=\sigma_{\mathcal{F}}^2-\sigma_{\mathcal{Q}}^2$。

8.9　说明均匀信源的 MMSE 量化器由公式(8.5.7)给出。

8.10 证明对于具有任意均值 η 和方差 σ^2 的信源，最优重建值和界限值可以由按照公式(8.5.17)对一个零均值和单位方差的信源所导出的最优重建值和界限值获得。

8.11 考虑一个具有概率密度函数 $p(f)=(\lambda/2)e^{-\lambda|f|}$ 的随机变量 $\mathcal{F}$。一个三级量化器定义为：

$$Q(f) = \begin{cases} b & f > a \\ 0 & -a \leqslant f \leqslant a \\ -b & f < -a \end{cases}$$

(a) 对于给定的 a，求 b 使得当失真测度是 MSE 时满足质心条件。

(b) 对于给定的 b，求 a 使之满足最邻近条件。

(c) 根据 λ，求 a,b 的最优集，使得两个条件都得到满足。导出最终的 MSE。

8.12 公式(8.5.12 及 8.5.13)中所给出的最优化条件是通过 MSE 最小化导出的。下面，我们考虑其他的失真测度。

(a) 如果失真准则是平均绝对误差(MAE)，也就是说 $D_q=E\{|F-Q(F)|\}$，导出一组类似的公式。证明判定区域 $\mathcal{B}_l=[b_{l-1},b_l]$ 的广义质心 $\mathbf{g}_l$ 是：

$$\int_{b_{l-1}}^{g_l} p(f \mid f \in \mathcal{B}_l)\, df = \int_{g_l}^{b_l} p(f \mid f \in \mathcal{B}_l)\, df$$

也就是说，g_l 按照概率相等地分割$\mathcal{B}_l$。这样的 g_l 也称为区域$\mathcal{B}_l$ 的中值。在这种失真测度下的最邻近条件的形式是什么？

(b) 对于具有拉普拉斯分布 $p(f)=\frac{1}{2}e^{-|f|}$ 的信源，确定在 MAE 准则下区域$\mathcal{B}_l$ 的广义质心。把这个质心与基于 MSE 准则 $D_q = E\{|\mathcal{F} - Q(\mathcal{F})|^2\}$所确定的质心相比较。

8.13 考虑离散矢量信源$\mathcal{F} = \{\mathbf{f}_k, k = 1,2,\cdots,K\}$的量化器，其中，$\mathbf{f}_k$ 是 N 维矢量。设两个矢量之间的失真测度是汉明距离，定义为：

$$d(\mathbf{f},\mathbf{g}) = \begin{cases} 0 & \mathbf{f} = \mathbf{g} \\ 1 & \text{其他} \end{cases}$$

假设要把这些矢量量化成 $L < K$ 个码字。

(a) 对于信源的一个给定的分割区域$\mathcal{B}_l = \{\mathbf{f}_k, k = 1,2,\cdots,K_l\}$，基于质心规则为这个区域确定一个新的码字。

(b) 对于给定的码字集 $C = \{\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_L\}$，基于最邻近规则确定一个新的信源分割。这个分割是惟一的吗？

(c) 从信源的一个任意的初始分割开始，需要多少次迭代才能达到一个局部最优的码书？

(提示：分别考虑两种情况：(1)当所有的训练矢量不相同时；(2)当一些训练矢量相同时。)

8.14 一个二维矢量量化器有两个码字：$\mathbf{g}_1 = [1/2, 1/2]^T$，$\mathbf{g}_2 = [-1/2, -1/2]^T$。设输入矢量 $\mathbf{f} = [f_1, f_2]^T$ 均匀分布在一个由 $-1 < f_1 < 1$ 和 $-1 < f_2 < 1$ 定义的正方形内。图示与两个码字有关的分割区域，并确定此量化器的 MSE(写出积分公式即可)。

8.15 一个有限区域上的二维点阵量化器由 $g_1 = m$ 和 $g_2 = m + 2n$ 定义的码字(g_1, g_2)组成，其中 m, n 是整数，使得(g_1, g_2)落入指定的区域。

(a) 画出原点附近的码字和分割区域。

(b) 设计一个算法，它为平面上的任意一个给定的输入点确定最接近的码字。

(c) 用(b)中的算法，求点(123.4,456.7)的最接近的码字。

(d) 如果输入均匀地分布在由所有分割区域所张成的空间，确定此量化器的 MSE。

8.16 基于由一个二维 i.i.d. 高斯信源生成的训练数据，写出实现广义劳埃德算法的 C 或 MATLAB代码。假设每个样点的两个分量是独立的，并且二者都是零均值和单位方差的。可以选择所用的样点数和码书的大小。如图 8.15 那样用二维图中的叉和圆圈标出每次迭代后的训练样点和码字。

8.17 重复习题 8.16，但假设两个矢量分量与相关系数 ρ 有关。比较用不同的 ρ 值所得的结果。

8.9 文献目录

[1] Berger, T. *Rate Distortion Theory*. Englewood Cliffs, NJ: Prentice Hall, 1971.

[2] Conway, J. H., and N. J. A. Sloane. Fast quantizing and decoding algorithms for lattice quantizers and codes. *IEEE Trans. Inform. Theory* (March 1982), IT-28:227–32.

[3] Cover, T. M., and J. A. Thomas. *Elements of Information Theory*. New York: John Wiley & Sons, 1991.

[4] Gersho, A., and R. M. Gray. *Vector Quantization and Signal Compression*. Boston: Kluwer Academic Press, 1992.

[5] Jayant, N. S., and P. Noll. *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.

[6] Linde, Y., A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Trans. Commun.* (Jan. 1980), COM-28:84–95.

[7] Lloyd, P. S. Least squares quantization in PCM. *IEEE Trans. Inform. Theory* (March 1982), IT-28:127–35. (Originally published as Bell Laborataries Technical Note, 1957.)

[8] MacQueen, J. Some methods for classification and analysis of multivariate observations. In *Fifth Berkeley Symposium on Math. Stat. and Prob.* (1967), 1:281–96.

[9] Sayood, K. *Introduction to Data Compression*. San Francisco: Morgan Kaufmann, 1996.

[10] Welch, T. A. A technique for high performance data compression, *IEEE Computer Magazine* (June 1984), 8–19.

[11] Witten, I. H., R. M. Neal, and J. G. Cleary. Arithmetic coding for data compression. *Commun. ACM* (1987), 30:520–40.

[12] Ziv, J., and A. Lempel. A universal algorithm for sequential data compression. *IEEE Trans. Inform. Theory* (May 1977), IT-23(3):337–43.

第9章　基于波形的视频编码

第8章介绍了基于波形和基于内容的视频编码的基本技术。本章将集中介绍基于波形的视频编码技术。首先,描述基于块的变换编码技术(9.1节),这种技术把每个像素块转换成一组不相关的系数,然后再量化编码。我们介绍重要的卡胡南-洛耶夫变换(KLT)以及它的近似——在大多数现代图像编码标准中使用的离散余弦变换(DCT)。采用另一种重要变换,即小波变换的编码方法将在第11章中描述可分级编码时介绍。9.2节介绍采用空间和时间预测的编码技术。这些技术利用相邻像素的空间相关性以及序列的相继帧之间的时间相关性来降低视频序列的码率。(空间相关性源于这样的事实:同一视频帧中相邻像素的彩色值通常都是平滑变化的;时间相关性是指这样的事实:序列的相继帧通常表现相同的物理场景,这个场景可能被相同物体所占据,而物体可能是移动的。)我们通过给出一个编码方案来总结本章,这个方案联合了变换和预测编码(9.3节),是已经在所有国际视频编码标准中采用的一种流行而有效的技术。

9.1　基于块的变换编码

已经证明变换编码对于压缩静止图像和视频帧特别有效。理想情况下,为了充分利用像素之间的空间相关性,变换应该应用于整幅图像或视频帧。但为了降低计算的复杂度,基于块的变换编码(把图像分成非重叠块并在每个块上应用变换)在实际中更为常用。本章只描述基于块的变换编码,集中介绍标准图像和视频编码器中所用的一类算法。对于变换编码更广泛的介绍,读者可以参考 Clark [6]。

我们从变换编码概述开始,概念性地介绍如何进行以及为什么要进行变换编码(9.1.1节)。然后正式定义一维和二维酉变换(9.1.2节及9.1.3节),并且作为例子定义DCT变换(9.1.4节)。9.1.5节讨论如何最佳地在变换系数之间分配比特,使得均方误差为最小。这一节也要导出标量量化变换编码的增益。这些结果是在假设任意的酉变换下获得的。9.1.6节导出一个变换要达到所有酉变换之间最高编码增益的必要条件,并且说明了KLT是这种意义上的最佳变换。然而,KLT取决于信号的统计特性而且很难计算;幸运的是,对于大多数图像信号,DCT是KLT的一个很好的近似。因此,基于DCT的编码已被所有的图像和视频编码标准所采用。9.1.7节给出基于DCT的图像编码算法,与JPEG标准所用的算法相同。用于视频编码的这个算法的修正在9.3.1节和第13章中介绍。在9.1.8节中我们以对矢量变换的简略介绍来结束本节。

9.1.1　概述

最流行的基于波形的图像编码方案之一是变换编码。在基于块的变换编码中,一个图像分解为非重叠的块,每个块变换成一组系数。这些系数用标量量化器分别量化。最后用可变长编码把量化的系数索引转换成二进制比特。在解码器中,通过逆变换由量化系数恢复图像块。图9.1给出了一个典型变换编码器的编码器和解码器的工作过程。

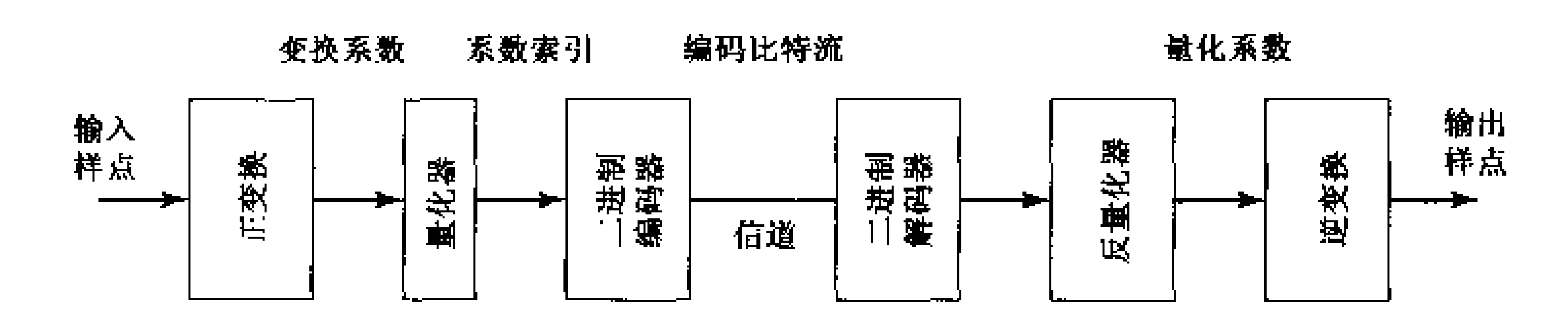


图 9.1　一个块变换编码系统的编码器和解码器的方框图

把变换看做到所选择的基函数上的投影　我们可以把变换过程认为是把一个图像块表示为一组基本图形(称为变换基函数)的线性组合,如图 9.2 所示。每个基图形的贡献是对应于那个变换基函数的变换系数。对于一个给定的图像块,导出变换系数的过程是正变换,而用变换系数重建图像块的过程是逆变换。

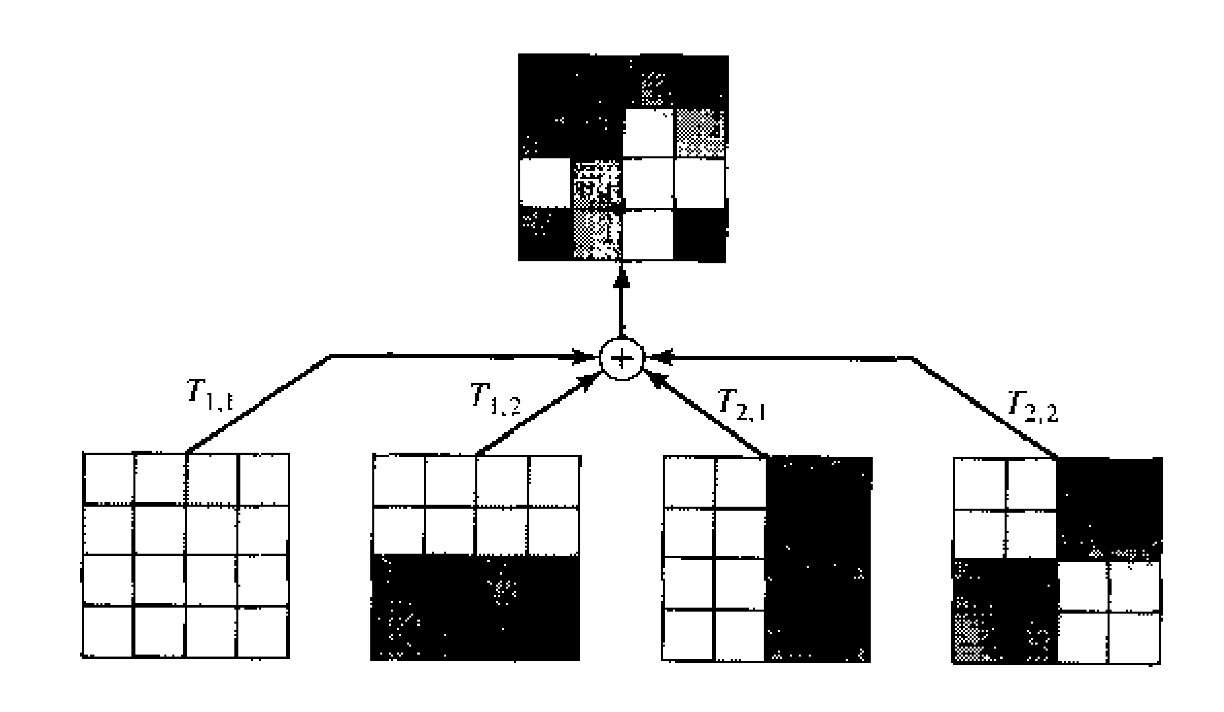


图 9.2　一个二维变换是图像在一组基本方块图形上的投影,每个基本图形都称为一个变换基

变换设计准则　显然,变换编码器的性能取决于所用的基函数。一个好的变换应该:(1)对欲量化的信号去相关,以便可以对各个值有效地使用标量量化而不会损失太多的编码效率(与矢量量化相比)。(2)把原始像素块的能量尽量压缩到少数的几个系数。后一个特性允许我们用几个具有大幅度的系数表示原始块。如 9.1.6 节将要说明的,在这些准则下最好的变换是 KLT。但是,因为 KLT 取决于信号的二阶统计特性且难以计算,所以实际中用固定的变换来近似 KLT。对于一般的图像信号,最近似 KLT 的变换是 DCT,因此在几乎所有的基于变换的图像编码器中都用 DCT。

变换编码和矢量量化　通常图像中的相邻像素之间是相关的,因此单独表示每个像素值是效率不高的。利用相邻像素之间的相关性的一种方法是用矢量量化把像素块一起量化,它用最接近原始块的一个典型的块图形来代替每个图像块。块越大,就越能充分利用像素之间的相关性,可以达到的压缩增益就越高。遗憾的是,如 8.6 节所述,搜索最佳匹配模式的复杂度也随块的大小而呈指数增长。变换编码是无需穷尽搜索而实现条件矢量量化器的一种途径。具体地说,这种量化器中的码字可以用量化的系数和基矢量进行线性组合来表示。因为每个系数取决于整个图像块,所以这些变换系数的标量量化实际上完成整个块的矢量量化,但降低了复杂度。

9.1.2 一维酉变换

一般的一维线性变换 如上所述，变换过程可以认为把图像块分解成一组基本的块图形。实际中，对于二维线性变换也是如此。在正式定义二维变换前，我们先定义一维变换。在这种情况下，一维信号被化分为长度为 N 的矢量，并把每个矢量表示为一组基矢量的线性组合。每个变换系数表示一个基矢量的贡献。从数学上讲，可以借助矩阵运算来获得这些系数。令 $\mathbf{s}=[s_1,s_2,\cdots,s_N]^T$ 表示由原始样点组成的矢量，而 $\mathbf{t}=[t_1,t_2,\cdots,t_N]^T$ 表示包含所有变换系数的矢量。进一步，令 $\mathbf{u}_k=[u_{k;1},u_{k;2},\cdots,u_{k;N}]^T$ 表示对应第 k 个变换基的矢量，而$[\mathbf{U}]=[\mathbf{u}_1,\mathbf{u}_2,\cdots,\mathbf{u}_N]$表示包含所有基矢量的矩阵。于是，表示样点矢量 $\mathbf{s}$ 的逆变换为：

$$\text{逆变换}:\mathbf{s} = \sum_{k\in\mathcal{N}} t_k\mathbf{u}_k = [\mathbf{U}]\mathbf{t} \tag{9.1.1}$$

其中$\mathcal{N}=\{1,2,\cdots,N\}$。为了能用 N 个基矢量表示任意的 N 维矢量，基矢量必须线性独立，在这种情况下矩阵$[\mathbf{U}]$必须可逆，变换系数由下式确定：

$$\text{正变换}:\mathbf{t} = [\mathbf{U}]^{-1}\mathbf{s} = [\mathbf{V}]\mathbf{s} \tag{9.1.2}$$

为了一般性，我们假设基矢量 $\mathbf{u}_k$ 可以包含复分量，从而 $\mathbf{u}_k$ 属于 N 维复空间C^N。在这种情况下，即使 $\mathbf{s}$ 是实矢量，$\mathbf{t}$ 一般都是复矢量。在以下的讨论中，我们假设 $\mathbf{s}$ 和 $\mathbf{t}$ 都属于C^N。

一维酉变换 上述讨论对变换基矢量未加任何约束。一种特殊类型的线性变换是酉变换，其中基矢量相互正交。

在正式定义酉变换之前，首先回顾一下复空间C^N 的内积和范数的定义。C^N 中的两个矢量 $\mathbf{s}_1$ 和 $\mathbf{s}_2$ 的内积定义为：

$$\langle\mathbf{s}_1,\mathbf{s}_2\rangle = \mathbf{s}_1^H\mathbf{s}_2 = \sum_{n\in\mathcal{N}} s_{1;n}^*s_{2;n} \tag{9.1.3}$$

其中上标 H 表示矢量转置后再对元素取共轭。如果$\langle\mathbf{s}_1,\mathbf{s}_2\rangle=0$，则称这两个矢量相互正交。$\mathbf{s}\in C^N$的范数 $\|\mathbf{s}\|$ 定义为：

$$\|\mathbf{s}\|^2 = \langle\mathbf{s},\mathbf{s}\rangle = \sum_{n\in\mathcal{N}} |\mathbf{s}_n|^2 \tag{9.1.4}$$

范数 $\|\mathbf{s}\|$ 表示 $\mathbf{s}$ 的长度或幅度，而范数的平方 $\|\mathbf{s}\|^2$ 表示 $\mathbf{s}$ 的能量。具有单位范数的矢量称为归一化矢量。如果两个矢量相互正交，并且每一个都是归一化的，则称它们为正交归一的。

如果公式(9.1.2)中定义的线性变换的基矢量相互正交，也就是说，如果

$$\langle\mathbf{u}_k,\mathbf{u}_l\rangle = \sum_{n\in\mathcal{N}} u_{k;n}^*u_{l;n} = \delta_{k,l} = \begin{cases}1 & k=l\\0 & k\neq l\end{cases} \tag{9.1.5}$$

那么矩阵$[\mathbf{U}]$满足：

$$[\mathbf{U}]^H[\mathbf{U}] = [\mathbf{U}][\mathbf{U}]^H = [\mathbf{I}]_N \tag{9.1.6}$$

其中$[\mathbf{I}]_N$ 表示一个 $N\times N$ 的单位矩阵。满足这个条件的矩阵称为酉矩阵。变换矩阵$[\mathbf{U}]$是酉矩阵的线性变换称为酉变换。在这种情况下$[\mathbf{U}]^{-1}=[\mathbf{U}]^H$，并且可以由 $\mathbf{t}=[\mathbf{U}]^H\mathbf{s}$ 或 $t_k=\langle\mathbf{u}_k,\mathbf{s}\rangle=\mathbf{u}_k^H\mathbf{s}$ 求出变换系数。也就是说，第 k 个变换系数 t_k 可以简单地由 $\mathbf{s}$ 与 $\mathbf{u}_k$ 的内积求得。

总而言之，一个酉变换对定义为：

$$\begin{aligned}&\text{正变换}:t_k = \langle\mathbf{u}_k,\mathbf{s}\rangle \text{ 或 } \mathbf{t} = [\mathbf{U}]^H\mathbf{s} = [\mathbf{V}]\mathbf{s}\\&\text{逆变换}:\mathbf{s} = \sum_{k\in\mathcal{N}} t_k\mathbf{u}_k = [\mathbf{U}]\mathbf{t} = [\mathbf{V}]^H\mathbf{t}\end{aligned} \tag{9.1.7}$$

通常，我们用正变换矩阵$[\mathbf{V}]=[\mathbf{U}]^H$来引用一个变换。变换的基矢量是$[\mathbf{V}]$中行矢量的共轭。

例 9.1　（一维 DFT）

回忆N点DFT定义为：

$$
\begin{aligned}
&\text{正变换}: t_k = \frac{1}{\sqrt{N}}\sum_{n=0}^{N-1} s_n e^{-j2\pi kn/N}, k = 0,1,\cdots,N-1 \\
&\text{逆变换}: s_n = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} t_k e^{j2\pi kn/N}, n = 0,1,\cdots,N-1
\end{aligned}
\tag{9.1.8}
$$

用已经描述过的概念，可以看出它是具有如下定义的基矢量的一个酉变换：

$$
u_{k;n} = \frac{1}{\sqrt{N}} e^{j2\pi\frac{kn}{N}}, n = 0,1,\cdots,N-1, k = 0,1,\cdots,N-1 \tag{9.1.9}
$$

或

$$
\mathbf{u}_k = \frac{1}{\sqrt{N}}[1 \ e^{j2\pi\frac{k}{N}} \ \cdots \ e^{j2\pi\frac{k(N-1)}{N}}]^T, k = 0,1,\cdots,N-1 \tag{9.1.10}
$$

注意，在这个例子中，我们按照习惯使索引k和n的取值范围为0到$N-1$，而非1到N。

酉变换的性质　到目前为止，我们假设$\mathbf{s}$是给定的样点矢量，而$\mathbf{t}$是它的变换。实际上，$\mathbf{s}$是随机矢量$\mathcal{S}=[\mathcal{S}_1,\mathcal{S}_2,\cdots,\mathcal{S}_N]^T$的一个实现，对应于从信源中抽取的任何$N$个样点。类似地，$\mathbf{t}$是随机矢量$\mathcal{T}=[\mathcal{T}_1,\mathcal{T}_2,\cdots,\mathcal{T}_N]^T=[\mathbf{V}]\mathcal{S}$的一个实现。在变换编码中，我们不仅对$\mathbf{s}$和$\mathbf{t}$之间的关系感兴趣，而且也对$\mathcal{S}$与$\mathcal{T}$的统计特性之间的关系感兴趣。对于酉变换，存在以下关系：

1. 均值矢量$\boldsymbol{\eta}_s=E\{\mathcal{S}\}$与$\boldsymbol{\eta}_t=E\{\mathcal{T}\}$的关系为：

$$
\boldsymbol{\eta}_t = [\mathbf{V}]\boldsymbol{\eta}_s, \boldsymbol{\eta}_s = [\mathbf{V}]^H\boldsymbol{\eta}_t \tag{9.1.11}
$$

协方差矩阵$[\mathbf{C}]_s=E\{(\mathcal{S}-\boldsymbol{\eta}_s)(\mathcal{S}-\boldsymbol{\eta}_s)^H\}$与$[\mathbf{C}]_t=E\{(\mathcal{T}-\boldsymbol{\eta}_t)(\mathcal{T}-\boldsymbol{\eta}_t)^H\}$之间的关系为：

$$
[\mathbf{C}]_t = [\mathbf{V}][\mathbf{C}]_s[\mathbf{V}]^H, [\mathbf{C}]_s = [\mathbf{V}]^H[\mathbf{C}]_t[\mathbf{V}] \tag{9.1.12}
$$

2. 变换矢量的总能量等于样点矢量的总能量。这对于给定的实现和全体的平均都是正确的。也就是说，

$$
\sum_{n\in\mathcal{N}} s_n^2 = \sum_{k\in\mathcal{N}} t_k^2 \tag{9.1.13}
$$

$$
\sum_{n\in\mathcal{N}} \sigma_{s,n}^2 = \sum_{k\in\mathcal{N}} \sigma_{t,k}^2 \tag{9.1.14}
$$

其中，$\sigma_{s,n}^2=E\{(\mathcal{S}_n-\eta_{s,n})^2\}$和$\sigma_{t,k}^2=E\{(\mathcal{T}_k-\eta_{t,k})^2\}$分别是$\mathcal{S}_n$和$\mathcal{T}_k$的方差。（这个性质等价于傅里叶变换的帕斯维尔定理。）

3. 设我们只用前$K<N$个系数来近似$\mathbf{s}$，近似矢量是$\hat{\mathbf{s}}_K=\sum_{k=1}^{K} t_k\mathbf{u}_k$，则近似误差信号是$\mathbf{e}_K=\mathbf{s}-\hat{\mathbf{s}}_K=\sum_{k=K+1}^{N} t_k\mathbf{u}_k$。对于给定的$\mathbf{s}$，近似误差的能量是：

$$
\|\mathbf{e}_K\|^2 = \sum_{n\in\mathcal{N}} e_n^2 = \sum_{k=K+1}^{N} t_k^2 \tag{9.1.15}
$$

$\mathcal{E}_K=\mathcal{S}-\hat{\mathcal{S}}_K$在$\mathcal{S}$全体上的方差是：

$$E\{\| \mathcal{E}_K \|^2\} = \sum_{n \notin \mathcal{N}} \sigma_{t,n}^2 = \sum_{k=K+1}^{N} \sigma_{t,k}^2 \tag{9.1.16}$$

因为所有系数的平方或方差的和是常数(公式(9.1.13)或(9.1.14)),如果我们选择具有最大值的 K 个系数来近似原始信号,则近似误差对于一个特殊的信号矢量是最小的。类似地,通过选择 K 个具有最大方差的系数可使平均近似误差最小。

以上关系式的证明是相当简单的,留做课后练习(习题 9.1)。

9.1.3 二维酉变换

上述讨论假设输入样点是来自一维信号。对于二维图像块,如果我们把 $\mathbf{s}$ 认为是按一维次序排列的一个图像块的样点所组成的矢量,并且把每个 $\mathbf{u}_k$ 认为是一个基图像按同样的一维次序排列的矢量表示,那么这些描述仍然适用。然而,我们也可以把一维变换的定义推广到二维。在这种情况下,一个 $M \times N$ 的图像块 $[\mathbf{S}] = [S_{m,n}]$ 被表示为 $M \times N$ 个基图像 $[\mathbf{U}]_{k,l} = [U_{k,l;m,n}]$, $k \in \mathcal{M}$, $l \in \mathcal{N}$ 的线性组合(见图 9.2),即:

$$[\mathbf{S}] = \sum_{k \in \mathcal{M}} \sum_{l \in \mathcal{N}} T_{k,l} [\mathbf{U}]_{k,l} \tag{9.1.17}$$

其中 $\mathcal{M} = \{1,2,\cdots,M\}$, $\mathcal{N} = \{1,2,\cdots,N\}$。

令 $C^{M \times N}$ 表示 $M \times N$ 维的复空间。令 $[\mathbf{S}]_1 = [S_{1;m,n}]$, $[\mathbf{S}]_2 = [S_{2;m,n}]$ 是该空间的两个矩阵。则这两个矩阵的内积定义为:

$$\langle [\mathbf{S}]_1, [\mathbf{S}]_2 \rangle = \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} S_{1;m,n}^* S_{2;m,n} \tag{9.1.18}$$

矩阵 $[\mathbf{S}] = [S_{m,n}]$ 的范数定义为:

$$\| [\mathbf{S}] \|^2 = \langle [\mathbf{S}], [\mathbf{S}] \rangle = \sum_{m \in \mathcal{M}} \sum_{n \in \mathcal{N}} | S_{m,n} |^2 \tag{9.1.19}$$

根据这些定义,一组基图像 $[\mathbf{U}]_{k,l}$ ($k \in \mathcal{M}$, $l \in \mathcal{N}$) 被称为相互正交,如果:

$$\langle [\mathbf{U}]_{k,l}, [\mathbf{U}]_{i,j} \rangle = \delta_{k,i} \delta_{l,j} = \begin{cases} 1, k = i, l = j \\ 0, 其他 \end{cases} \tag{9.1.20}$$

对于一组正交的基图像,可以很容易地由 $T_{k,l} = \langle [\mathbf{U}]_{k,l}, [\mathbf{S}] \rangle$ 求出变换系数。一个 M × N 的二维酉变换的正变换与逆变换的关系是:

$$\begin{aligned} &正变换: T_{k,l} = \langle [\mathbf{U}]_{k,l}, [\mathbf{S}] \rangle \\ &逆变换: [\mathbf{S}] = \sum_{k \in \mathcal{M}} \sum_{l \in \mathcal{N}} T_{k,l} [\mathbf{U}]_{k,l} \end{aligned} \tag{9.1.21}$$

例 9.2 (二维 DFT)

一个大小为 $M \times N$ 的二维 DFT 定义为:

$$\begin{aligned} &正变换: T_{k,l} = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} S_{m,n} e^{-j2\pi\left(\frac{km}{M} + \frac{ln}{N}\right)} \\ &k = 0,1,\cdots,M-1, l = 0,1,\cdots,N-1; \\ &逆变换: S_{m,n} = \frac{1}{\sqrt{MN}} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} T_{k,l} e^{j2\pi\left(\frac{km}{M} + \frac{ln}{N}\right)} \\ &m = 0,1,\cdots,M-1, n = 0,1,\cdots,N-1 \end{aligned} \tag{9.1.22}$$

可以看出这是一种二维酉变换，其基图像定义为：

$$U_{k,l;m,n} = \frac{1}{\sqrt{MN}} e^{j2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)}, m,k = 0,1,\cdots,M-1, n,l = 0,1,\cdots,N-1 \quad (9.1.23)$$

可分离的变换　一类特别重要的二维变换是它的每一个基图像是两个一维变换的基矢量的外积。事实上，使用的所有二维变换都属于此类。特别地，令 $\mathbf{h}_k(k\in\mathcal{M})$和 $\mathbf{g}_l(l\in\mathcal{N})$分别表示$C^M$ 和C^N 中的基矢量，则我们可以由下式构成$C^{M\times N}$中的基图像：

$$[\mathbf{U}]_{k,l} = \mathbf{h}_k\mathbf{g}_l^T \text{或} U_{k,l;m,n} = h_{k;m}g_{l;n}, k\in\mathcal{M}, l\in\mathcal{N} \quad (9.1.24)$$

可以证明只要 $\mathbf{h}_k(k\in\mathcal{M})$和 $\mathbf{g}_l(l\in\mathcal{N})$分别形成$C^M$ 和C^N 中的正交基组，则$[\mathbf{U}]_{k,l}(k\in\mathcal{M}, l\in\mathcal{N})$就将形成$C^{M\times N}$中的正交基组。

例 9.3　（可分离变换的二维 DFT）

回忆 $M\times N$ 点的二维 DFT 的基图像是：

$$U_{k,l;m,n} = \frac{1}{\sqrt{MN}} e^{j2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)}$$

而 M 点 DFT 和 N 点 DFT 的基矢量分别是：

$$h_{k;m} = \frac{1}{\sqrt{M}} e^{j2\pi\frac{km}{M}}, g_{l;n} = \frac{1}{\sqrt{N}} e^{j2\pi\frac{ln}{N}}$$

很明显，$U_{k,l;m,n} = h_{k;m}g_{l;n}$。因此，$M\times N$ 点的二维 DFT 是由一个 M 点的一维 DFT 和一个 N 点的一维 DFT 构成的可分离的变换。

当变换可分离时，我们可以先对每行用基矩阵$[\mathbf{G}] = [\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_N]$完成一维变换，然后再对中间图像的每列用基矩阵$[\mathbf{H}] = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_M]$执行一维变换。这个操作可以表示为：

$$\begin{aligned} &\text{正变换}:[\mathbf{T}] = [\mathbf{H}]^H[\mathbf{S}][\mathbf{G}] \\ &\text{逆变换}:[\mathbf{S}] = [\mathbf{H}][\mathbf{S}][\mathbf{G}]^H \end{aligned} \quad (9.1.25)$$

当二维变换可分离时，它的等效一维变换表示的变换矩阵将是矩阵$[\mathbf{G}]$和$[\mathbf{H}]$的克罗耐克(Kronecker)积。对于这个课题更详尽的讨论，包括可分离变换的其他性质，见[18,5.2 节]。

可分离变换在计算上的节省　为确定一个 $M\times N$ 变换的每个系数，一般需要进行 $M\times N$ 次运算，为了确定所有的 $M\times N$ 个系数，计算总数为 M^2N^2，当 $M=N$ 时为 N^4。另一方面，当变换是可分离的，我们首先沿 M 行的每一行计算 N 点一维变换，每行需要 N^2 次运算。再沿 N 列的每一列计算 M 点一维变换，每列需要 M^2 次运算。总的计算次数是 MN^2+NM^2，如果 $M=N$，就是 $2N^3$。当 N 较大时，从 N^4 到 $2N^3$ 的减小是显著的。

如果存在计算一维变换的快速算法，那么可以获得进一步的节省。例如，如果 N 点一维变换需要 $N\log_2 N$ 次运算，那么 $N\times N$ 的二维可分离变换可以用 $2N^2\log_2 N$ 次运算完成。

9.1.4　离散余弦变换

除 DFT 外，还已经为信号分析开发出许多其他的变换，包括离散余弦变换(DCT)、离散正弦变换(DST)、阿达玛(Hadamard)变换、沃尔什(Walsh)变换、哈尔(Haar)变换以及斜变换(这些变换的基矢量的定义见参考文献[12]或[18])。所有这些变换起初都是以一维形式定义的，可以用来构造二维可分离的变换。DFT 最广泛地应用于离散信号的频域分析。然而，已发现 DCT

对信源编码特别是图像编码更有用。因此,本节介绍 DCT,并给出应用它来表示图像的例子。

一维 N 点 DCT 的基矢量定义为:

$$u_{k;n} = \alpha(k)\cos\frac{(2n+1)k\pi}{2N}, n = 0,1,\cdots,N-1 \tag{9.1.26}$$

其中

$$\alpha(k) = \begin{cases} \sqrt{\dfrac{1}{N}} & k = 0 \qquad (9.1.27) \\ \sqrt{\dfrac{2}{N}} & k = 1,2,\cdots,N-1 \qquad (9.1.28) \end{cases}$$

正变换和逆变换描述为:

$$t_k = \sum_{n=0}^{N-1} u_{k;n}^* s_n = \alpha(k)\sum_{n=0}^{N-1} s_n \cos\frac{(2n+1)k\pi}{2N} \tag{9.1.29}$$

$$s_n = \sum_{k=0}^{N-1} u_{k;n} t_k = \sum_{k=0}^{N-1} \alpha(k) t_k \cos\frac{(2n+1)k\pi}{2N} \tag{9.1.30}$$

与 DFT 一样,我们根据习惯对 k 和 n 取 0 到 $N-1$ 的索引范围。

注意,DCT 基矢量是实矢量,随频率的增加以正弦图形变化。每个 DCT 系数确定正弦图形在实际信号的一个特定频率上的贡献。最低的系数称为 *DC* 系数,表示信号的平均值。其他的系数(称为 *AC* 系数)与递增的较高的频率相联系。

二维 $M \times N$ 点的 DCT 是由一维 M 点 DCT 基和 N 点 DCT 基构成的。也就是说,每个 $M \times N$ 的基图像是 M 点 DCT 基矢量与 N 点 DCT 基矢量的外积。图 9.3 给出了对应 8×8 DCT 的基图像。为了获得一个图像块的二维 DCT,首先可以对图像块的每行运用对应的一维 DCT,然后再对经行变换的块的每列应用一维 DCT。存在几种计算 N 点一维 DCT 的快速算法,只需 $N\log_2 N$ 次运算[4,21]。

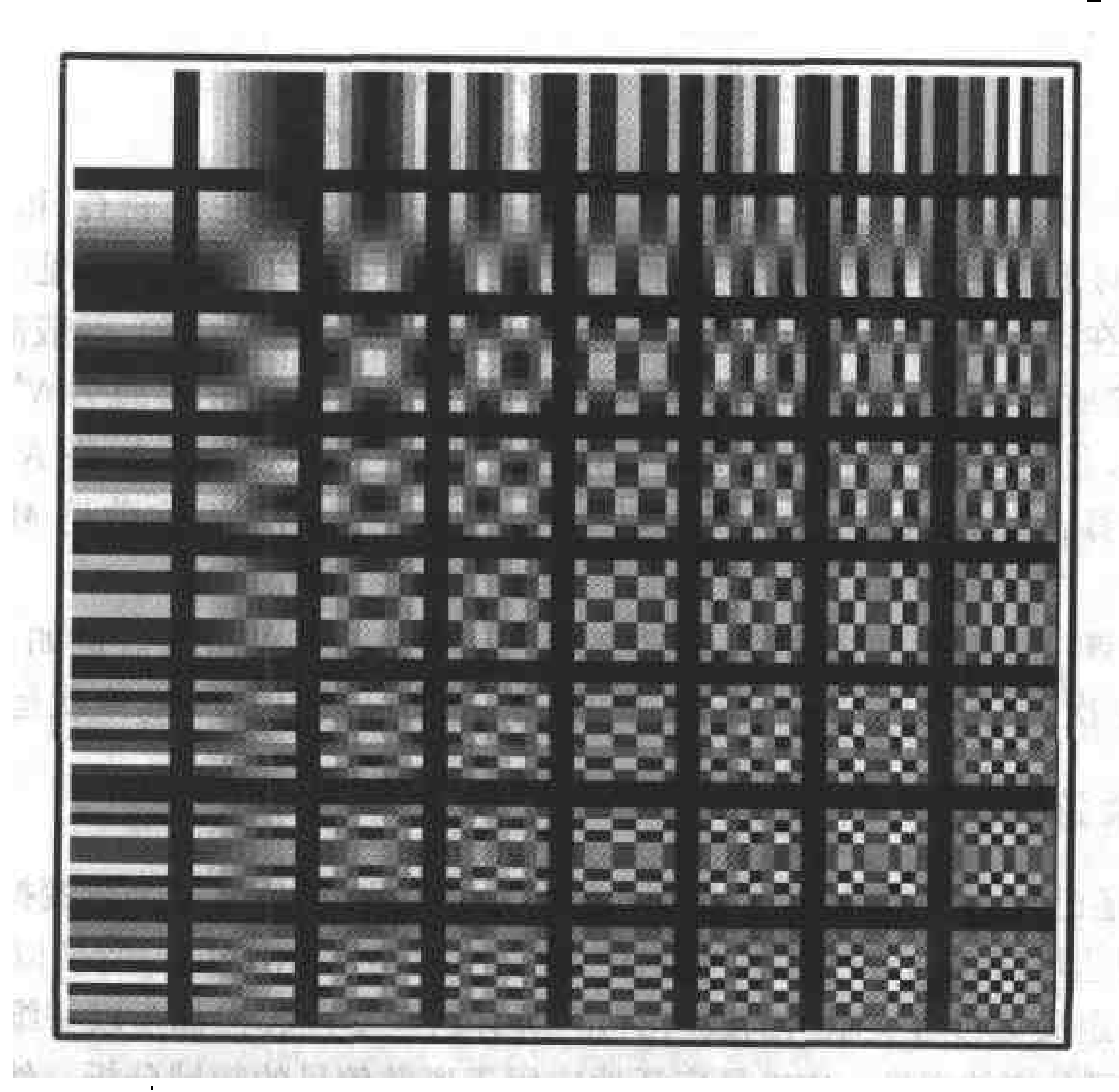


图 9.3　8×8 DCT 变换的基图像

DCT 能够很好地适用于图像编码,其原因是图像块经常可以用几个低频 DCT 系数表示。这是因为图像中的强度值通常是平滑变化的,高频部分只在边缘附近存在。图 9.4 给出了一个测试图像的 DCT 系数的能量(即方差)分布。用"Z"形扫描(在 9.1.7 节描述)排序系数,以便使垂直和水平方向上的低索引的系数放在前面。可以看出 DCT 系数的能量随频率索引的增加而迅速下降。图 9.5 示出了使用四种不同数目的 DCT 系数的测试图像的近似。可以看到,只用每个块的 64 个系数中的 16 个系数,我们就已经可以相当好地表示原始图像了。

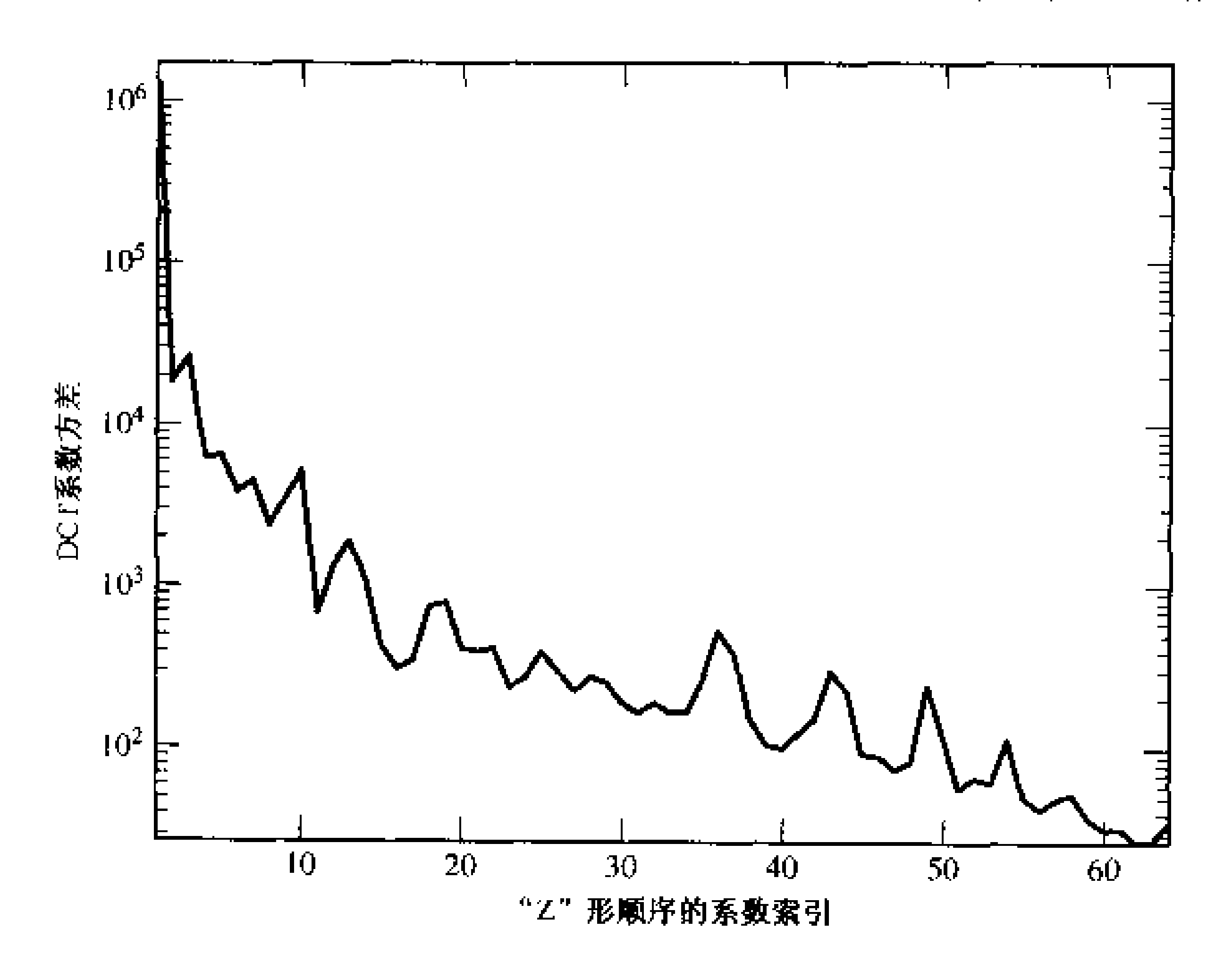


图 9.4　测试图像"花"的 8×8 DCT 系数的能量分布

9.1.5　比特分配和变换编码增益

在本节中,我们考虑变换编码器的率失真性能。尽管我们对二维变换最感兴趣,但我们在分析中将使用一维变换表示。因为任何二维信号都可以映射到一维,然后再变换,所以这种选择不会限制分析结果的可应用性。我们将假设变换是任意的一维酉变换。进一步假设分别量化变换系数,每个系数采用基于该系数概率密度函数的最佳的标量量化器;并且用固定长编码把量化系数转化为比特。我们的目的是要导出在给定的矢量中,每个样点的平均失真与每个样点的平均比特率之间的关系。我们也将把变换编码可以达到的性能与对每个原始样点进行标量量化的 PCM 系统的性能相比较。

样点失真与系数失真之间的关系　首先,我们必须建立重建样点的量化误差与变换系数的量化误差之间的关系。在本节中将无例外地用 MSE 作为失真准则。如前所述,我们用 $\mathcal{S}_n$ 和 $\mathcal{T}_k$ 分别表示对应于 s_n 和 t_k 的随机变量,用 $\boldsymbol{\mathcal{S}}$ 和 $\boldsymbol{\mathcal{T}}$ 分别表示对应于 $\mathbf{s}$ 和 $\mathbf{t}$ 随机矢量。除此之外,用 $\hat{s}_n, \hat{\mathbf{s}}, \hat{\mathcal{S}}_n, \hat{\boldsymbol{\mathcal{S}}}$ 表示 $s_n, \mathbf{s}, \mathcal{S}_n, \boldsymbol{\mathcal{S}}$ 的量化形式,用 $\hat{t}_k, \hat{\mathbf{t}}, \hat{\mathcal{T}}_k, \hat{\boldsymbol{\mathcal{T}}}$ 表示 $t_k, \mathbf{t}, \mathcal{T}_k, \boldsymbol{\mathcal{T}}$ 的量化形式。原始样点与重建样点之间的每个样点的 MSE 是:

$$D_s = \frac{1}{N}E\{\| \boldsymbol{\mathcal{S}} - \hat{\boldsymbol{\mathcal{S}}} \|^2\} = \frac{1}{N}\sum_{n \in \mathcal{N}} D_{s,n} \tag{9.1.31}$$



图 9.5 用不同数目的系数重建的图像“花”。左上图是每个块用 64 个系数的原始图像；右上图用 16 个系数；左下图用 8 个系数；而右下图用 4 个系数。DCT 系数是以“Z”形顺序排序的

其中

$$D_{s,n} = E\{(\mathcal{S}_n - \hat{\mathcal{S}}_n)^2\} \tag{9.1.32}$$

类似地，原始系数与量化系数之间的每个系数的 MSE 是：

$$D_t = \frac{1}{N}E\{\| \mathcal{T} - \hat{\mathcal{T}} \|^2\} = \frac{1}{N}\sum_{k\in\mathcal{N}} D_{t,k} \tag{9.1.33}$$

其中

$$D_{t,k} = E\{(\mathcal{T}_k - \hat{\mathcal{T}}_k)^2\} \tag{9.1.34}$$

使用关系

$$\mathcal{S} = [\mathbf{V}]^H \mathcal{T} \text{ 和 } \hat{\mathcal{S}} = [\mathbf{V}]^H \hat{\mathcal{T}} \tag{9.1.35}$$

我们有：

$$\begin{aligned} D_s &= \frac{1}{N}E\{\| \mathcal{S} - \hat{\mathcal{S}} \|^2\} = \frac{1}{N}E\{\| [\mathbf{V}]^H(\mathcal{T} - \hat{\mathcal{T}}) \|^2\} \\ &= \frac{1}{N}E\{(\mathcal{T} - \hat{\mathcal{T}})^H[\mathbf{V}][\mathbf{V}]^H(\mathcal{T} - \hat{\mathcal{T}})\} = \frac{1}{N}E\{\| \mathcal{T} - \hat{\mathcal{T}} \|^2\} = D_t \end{aligned}$$

其中第 4 个等式是根据[V]的酉性质。这个结果告诉我们样点域中的平均量化误差等于变换域中的平均量化误差。

令 $\sigma_{t,k}^2$ 表示系数 $\mathcal{T}_k$ 的方差，并且令 R_k 和 $D_{t,k}$ 表示 $\mathcal{T}_k$ 的比特率和失真。由公式(8.5.18)，当 R_k 足够高时，我们有：

$$D_{t,k}(R_k) = \epsilon_{t,k}^2 \sigma_{t,k}^2 2^{-2R_k} \tag{9.1.36}$$

其中$\epsilon_{t,k}^2$取决于$\mathcal{T}_k$的概率密度函数。因此，使用变换编码的平均失真是：

$$D_{\mathrm{TC}} = D_s = D_t = \frac{1}{N}\sum_{k\in\mathcal{N}} \epsilon_{t,k}^2 \sigma_{t,k}^2 2^{-2R_k} \tag{9.1.37}$$

最佳比特分配　给定期望的平均比特率 R，问题是如何在 N 个系数之间分配总的比特数 RN，使得公式(9.1.37)中的误差最小化。用拉格朗日乘子法，这个约束的最小化问题可以转化为使下式为最小：

$$J(R_k, \forall k \in \mathcal{N}) = \sum_{k\in\mathcal{N}} \epsilon_{t,k}^2 \sigma_{t,k}^2 2^{-2R_k} + \lambda\left(\sum_{k\in\mathcal{N}} R_k - RN\right) \tag{9.1.38}$$

其中必须选择 λ 以满足

$$\sum_{k\in\mathcal{N}} R_k = RN \tag{9.1.39}$$

如果我们令$(\partial J/\partial R_k)=0$，可得：

$$\frac{\partial D_{t,k}}{\partial R_k} = -2\ln 2 D_{t,k} = -(2\ln 2)\epsilon_{t,k}^2 \sigma_{t,k}^2 2^{-2R_k} = -\lambda, \quad \forall k \in \mathcal{N} \tag{9.1.40}$$

为了确定 λ，对所有的 k 取上述公式的乘积，得到：

$$\lambda^N = (2\ln 2)^N\left(\prod_k \epsilon_{t,k}^2 \sigma_{t,k}^2\right) 2^{-2\sum_k R_k} = (2\ln 2)^N\left(\prod_k \epsilon_{t,k}^2 \sigma_{t,k}^2\right) 2^{-2NR} \tag{9.1.41}$$

或

$$\lambda = (2\ln 2)\left(\prod_k \epsilon_{t,k}^2 \sigma_{t,k}^2\right)^{1/N} 2^{-2R} \tag{9.1.42}$$

把它代入公式(9.1.40)得到：

$$R_k = R + \frac{1}{2}\log_2 \frac{\epsilon_{t,k}^2 \sigma_{t,k}^2}{\left(\prod_k \epsilon_{t,k}^2 \sigma_{t,k}^2\right)^{1/N}} \tag{9.1.43}$$

通过这样的比特分配，所有系数造成的失真都相等，也就是说，

$$D_{\mathrm{TC}} = D_t = D_{t,k} = \left(\prod_k \epsilon_{t,k}^2 \sigma_{t,k}^2\right)^{1/N} 2^{-2R} \tag{9.1.44}$$

公式(9.1.43)中给出的解意味着方差较大的系数应该给予较多的比特，而方差较小的系数应该给予较少的比特。最佳分配使得所有系数具有相同的量化误差。

注意，对于具有足够小方差的系数，按照公式(9.1.43)所确定的码率可能是负的。在实际中，这个系数的码率必须置为零，相当于根本不对此系数进行编码。必须减少分配给其他系数的比特以满足原始的平均比特率规定。对于明确限制 R_k 是非负的或非负整数的比特分配算法，见参考文献[34,31]。

变换编码相对于 PCM 的增益　假设原始样点来源于平稳信源，从而每个样点都有相同的方差 σ_s^2。如用 PCM 编码此信源，也就是说把最佳标量量化应用于分离的各个样点，那么失真与码率的关系为：

$$D_{\mathrm{PCM}} = D_{s,n} = \epsilon_s^2 \sigma_s^2 2^{-2R} \tag{9.1.45}$$

其中，ϵ_s^2 取决于信源样点的概率密度函数。

变换编码的性能测度是相对于 PCM 的变换编码增益，定义为：

$$G_{TC}=\frac{D_{PCM}}{D_{TC}} \tag{9.1.46}$$

把公式(9.1.45)和(9.1.44)代入公式(9.1.46)，我们有：

$$G_{TC}=\frac{\epsilon_s^2\sigma_s^2}{(\prod_k \epsilon_{t,k}^2\sigma_{t,k}^2)^{1/N}}=\frac{\epsilon_s^2}{(\prod_k \epsilon_{t,k}^2)^{1/N}}\frac{\frac{1}{N}\sum\sigma_{t,k}^2}{(\prod_k\sigma_{t,k}^2)^{1/N}} \tag{9.1.47}$$

其中第二个等式是由公式(9.1.14)中所给出的酉变换的能量保持性质和 $\sigma_{s,n}^2=\sigma_s^2$ 而得到的。因此，变换编码增益正比于变换系数方差的算术平均与这些方差的几何平均之比。众所周知，对于任意一组值，算术平均等于或大于几何平均，仅当所有值都相等时这两个平均才相等。一般地，我们也有$\epsilon_s^2\geqslant(\prod_k\epsilon_{t,k}^2)^{1/N}$。所以在相同平均比特率下，$G_{TC}\geqslant1$ 或 $D_{s,TC}\leqslant D_{s,PCM}$。系数的方差值越不平衡，它们的几何平均就越小，编码增益就越高。

当信源是高斯型时，每个样点是高斯型的。因为每个变换系数是样点的线性组合，所以它也符合高斯分布。在这种情况下，$\epsilon_s^2=\epsilon_{t,k}^2$，因此：

$$G_{TC,Gaussian}=\frac{\sigma_s^2}{(\prod_k\sigma_{t,k}^2)^{1/N}}=\frac{\frac{1}{N}\sum\sigma_{t,k}^2}{(\prod_k\sigma_{t,k}^2)^{1/N}} \tag{9.1.48}$$

例 9.4 考虑对一幅图像中每个 2×2 的块用 2×2 DCT 进行变换编码。假设图像是方差为 σ_s^2 的平稳高斯过程，且水平方向、垂直方向以及对角线上的相邻像素之间的相关系数分别是 ρ_h，ρ_v 和 ρ_d，如图 9.6 所示。我们要对于给定的平均比特率 R 确定最佳的比特分配以及相应的失真，把这个编码器与用最佳标量量化器直接量化每个样点的 PCM 编码器进行比较，且给出对于 $\sigma_s^2=1$，$\rho_h=\rho_v=\rho=0.95$，$\rho_d=\rho^2=0.9025$ 和 $R=2$ 这一特殊情况下的数字结果。

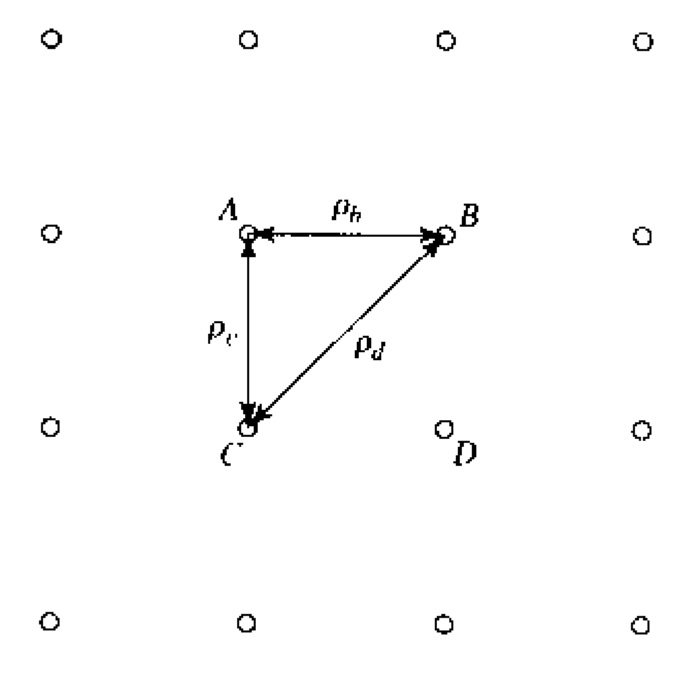


图 9.6 例 9.4～9.6 中所考虑的图像源

为求解此问题，把 2×2 个像素的数组排列成四维矢量，并把 2×2 的二维 DCT 当做 4 点的一维 DCT。用一维排序 $\mathbf{s}=[A,B,C,D]^T$，假设 $\mathbf{s}$ 是零均值的，则 $\mathbf{s}$ 的协方差矩阵是：

$$[\mathbf{C}]_s = E\left\{\begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix}[A\ B\ C\ D]\right\} = \begin{bmatrix} C_{AA} & C_{AB} & C_{AC} & C_{AD} \\ C_{BA} & C_{BB} & C_{BC} & C_{BD} \\ C_{CA} & C_{CB} & C_{CC} & C_{CD} \\ C_{DA} & C_{DB} & C_{DC} & C_{DD} \end{bmatrix}$$

$$= \sigma_s^2 \begin{bmatrix} 1 & \rho_h & \rho_v & \rho_d \\ \rho_h & 1 & \rho_d & \rho_v \\ \rho_v & \rho_d & 1 & \rho_h \\ \rho_d & \rho_v & \rho_h & 1 \end{bmatrix} \tag{9.1.49}$$

2×2 DCT 的基图像是：

$$\frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \frac{1}{2}\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}, \frac{1}{2}\begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}, \frac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{9.1.50}$$

把每个基图像重排成矢量，等效的一维逆变换矩阵是：

$$[\mathbf{U}] = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{9.1.51}$$

用公式(9.1.12)中的性质可以得到变换矢量$[\mathbf{C}]_t$ 的协方差矩阵，变换系数的方差是$[\mathbf{C}]_t$ 中的对角元素。在 $\rho_h = \rho_v = \rho = 0.95, \rho_d = \rho^2 = 0.9025$ 和 $\sigma_s^2 = 1$ 的特殊情况下，系数的方差是：

$$\sigma_{t,k}^2 = \{(1+\rho)^2, (1-\rho^2), (1-\rho^2), (1-\rho)^2\}\sigma_s^2 \tag{9.1.52}$$

$$= \{3.8025, 0.0975, 0.0975, 0.0025\} \tag{9.1.53}$$

系数方差的几何平均是：

$$\sigma_t^2 = \left(\prod_k \sigma_{t,k}^2\right)^{1/4} = (1-\rho^2)\sigma_s^2 = 0.0975 \tag{9.1.54}$$

如果所期望的比特率为 $R = 2$ bpp（比特/像素），那么应用公式(9.1.43)，4 个系数之间的最佳比特分配是：

$$R_k = \{4.64, 2, 2, -0.64\} \tag{9.1.55}$$

应用公式(9.1.44)和$\epsilon_{t,k}^2 = \epsilon_{\text{Gaussian}}^2$，每个像素的平均失真是：

$$D_{\text{TC}} = \epsilon_{\text{Gaussian}}^2 (1-\rho^2)\sigma_s^2 2^{-2R} = 0.0061\,\epsilon_{\text{Gaussian}}^2 \tag{9.1.56}$$

如果直接量化每个像素，那么应用公式(9.1.45)，平均失真（假设用最佳量化器）是：

$$D_{\text{PCM}} = \epsilon_{\text{Gaussian}}^2 \sigma_s^2 2^{-2R} = 0.0625\,\epsilon_{\text{Gaussian}}^2 \tag{9.1.57}$$

应用公式(9.1.48)，变换编码增益是：

$$G_{\text{TC}} = \frac{\sigma_s^2}{\sigma_t^2} = \frac{1}{1-\rho^2} = 10.25 \tag{9.1.58}$$

注意，公式(9.1.55)中的最佳比特分配假设可以用负数个比特对变量编码。实际中，必须设 $R_4 = 0$，并在剩余的三个变量之间重新分配总的比特数。简单的解决办法是令 $R_1 = 4$，从而修改后的比特分配是：

$$R_k = \{4, 2, 2, 0\} \tag{9.1.59}$$

在这种情况下的平均失真是：

$$D_{TC} = \frac{1}{4}\sum_{k} \epsilon^2_{Gaussian}\sigma^2_{t,k}2^{-2R_k} = 0.0074\epsilon^2_{Gaussian} \tag{9.1.60}$$

尽管这个失真比公式(9.1.56)指出的高，它仍比 PCM 可以达到的值低得多。

9.1.6 最佳变换设计和 KLT

正如我们刚刚说明的，通过最佳比特分配和最佳标量量化，变换编码产生比 PCM 方法低的平均失真。这对于任何酉变换都是正确的。那么自然会问是否存在使失真 D_s 最小(或等效地，使增益 G_{TC}最大)的最佳变换。因为对于一个给定的信源$\epsilon^2_s\sigma^2_s$ 是固定的，所以变换应该在所有的 $N\times N$ 酉变换中产生最小的几何平均$(\prod_k \epsilon^2_{t,k}\sigma^2_{t,k})^{1/N}$。幸运的是，如果信源是高斯型的，回答是肯定的，这样的最佳变换就是 KLT。

KLT 基的构造　对于 KLT，基矢量是基于原始信号$[\mathbf{C}]_s$ 的协方差矩阵设计的。令 λ_k 和 ϕ_k 分别表示$[\mathbf{C}]_s$ 的第 k 个特征值和归一化特征矢量。则它们满足：

$$[\mathbf{C}]_s\phi_k = \lambda_k\phi_k,\quad 其中\langle\phi_k,\phi_l\rangle = \delta_{k,l}$$

对于一个给定的协方差矩阵$[\mathbf{C}]_s$，KLT 是用 ϕ_k 作为基矢量的酉变换。相应的逆变换矩阵是：

$$[\mathbf{\Phi}] = [\phi_0,\phi_1,\cdots,\phi_{N-1}] \tag{9.1.61}$$

用$[\mathbf{\Phi}]^H$ 代替公式(9.1.12)中的$[\mathbf{V}]$，我们得到：

$$[\mathbf{C}]_t = [\mathbf{\Phi}]^H[\mathbf{C}]_s[\mathbf{\Phi}] = \begin{bmatrix}\phi_1^H\\ \phi_2^H\\ \cdots\\ \phi_N^H\end{bmatrix}[\mathbf{C}]_s[\phi_1,\phi_2,\cdots,\phi_N] \tag{9.1.62}$$

$$= \begin{bmatrix}\phi_1^H\\ \phi_2^H\\ \cdots\\ \phi_N^H\end{bmatrix}[\lambda_1\phi_1,\lambda_2\phi_2,\cdots,\lambda_N\phi_N] = \begin{bmatrix}\lambda_1 & 0 & \cdots & 0\\ 0 & \lambda_2 & \cdots & 0\\ \cdots & \cdots & \cdots & \cdots\\ 0 & 0 & \cdots & \lambda_N\end{bmatrix} \tag{9.1.63}$$

也就是说，KLT 使$[\mathbf{C}]_s$ 对角化。而且第 k 个变换系数的方差是$\sigma_k^2 = \lambda_k$。

为了证明 KLT 将使编码失真最小，可以利用不等式：

$$\det[\mathbf{C}]_t \leqslant \prod_{k\in\mathcal{N}} \sigma^2_{t,k} \tag{9.1.64}$$

此不等式适用于对角项为 $\sigma^2_{t,k}$的任何协方差矩阵$[\mathbf{C}]$。另一方面，由公式(9.1.12)，对于所有的酉变换有：

$$\det[\mathbf{C}]_t = \det[\mathbf{C}]_s \tag{9.1.65}$$

因此，对于任何酉变换，我们有：

$$\prod_{k\in\mathcal{N}} \sigma^2_{t,k} \geqslant \det[\mathbf{C}]_s \tag{9.1.66}$$

但对于 KLT，我们有：

$$\prod_{k\in\mathcal{N}} \sigma^2_{t,k} = \det[\mathbf{C}]_t = \det[\mathbf{C}]_s \tag{9.1.67}$$

这样，在任何酉变换中，KLT 产生变换系数方差的最小几何平均。因此，如果信源是高斯型的，

则它使变换编码增益最大。即使对于非高斯型的信源，通常ϵ^2 因子的影响可以忽略，从而可以认为 KLT 是最佳变换。

KLT 的另一个性质是它在所有的酉变换中用 $k<N$ 个系数得到最小的近似误差，这里误差的定义与公式(9.1.16)中相同。换言之，我们说 KLT 在所有变换中具有最高的能量紧缩能力。这是因为近似误差与系数方差的几何平均有直接的联系。几何平均越低，这些方差的分布就越不均匀，因此，能够紧缩到固定数目系数上的能量就越多。

对于 KLT 和高斯信源，我们有：

$$D_{\mathrm{TC}} = \epsilon^2_{\mathrm{Gaussian}}(\det[\mathbf{C}]_s)^{1/N}2^{-2R} \tag{9.1.68}$$

回想相关高斯矢量信源的 RD 界限由公式(8.3.26)给出。因此，由变换编码可以达到的运算 RD 函数超过了由常数因子$\epsilon^2_{\mathrm{Gaussian}}=2.71$ 可达到的界限。

由公式(9.1.47)和(9.1.65)，用 KLT 的编码增益是：

$$G_{\mathrm{TC,KLT}} = \frac{\epsilon_s^2}{(\prod_k \epsilon_k^2)^{1/N}}\frac{\sigma_s^2}{(\prod_k \lambda_k)^{1/N}} = \frac{\epsilon_s^2}{(\prod_k \epsilon_k^2)^{1/N}}\frac{\sigma_s^2}{(\det[\mathbf{C}]_s)^{1/N}} \tag{9.1.69}$$

例 9.5　再一次考虑图 9.6 中的图像信源。不用 DCT，而是采用 KLT。由任何 2×2 个像素组成的四维矢量的协方差矩阵$[\mathbf{C}]_s$ 由公式(9.1.49)给出。为了确定 KLT 的变换矩阵，我们需要确定$[\mathbf{C}]_s$ 的特征值和特征矢量。特征值由求解下式确定：

$$\det([\mathbf{C}]_s - \lambda[\mathbf{I}]) = 0 \tag{9.1.70}$$

对于每个特征值 λ_k，我们可以通过求下式确定相应的特征矢量 ϕ_k：

$$([\mathbf{C}]_s - \lambda[\mathbf{I}])\phi_k = \mathbf{0} \tag{9.1.71}$$

其中约束 $\|\phi_k\|^2=1$。对于 $\rho_h=\rho_d=\rho,\rho_d=\rho^2$ 的特殊情况，所得到的特征值是：

$$\lambda_k = \{(1+\rho)^2,(1-\rho^2),(1-\rho^2),(1-\rho)^2\}\sigma_s^2$$

把此结果与公式(9.1.52)比较，可以看出这种情况下的特征值与例 9.4 中用 DCT 所得到的方差相同。从而由其特征矢量组成的 KLT 矩阵与公式(9.1.51)给出的 DCT 变换矩阵相同[①]。很容易证明 $\det[\mathbf{C}]_s=\prod_k\lambda_k$。

在这个特例中，对于 ρ 的任何值 DCT 都等于 KLT。可以证明，对于任意的一阶马尔可夫过程，它的 KLT 基函数非常类似 DCT 的基函数[1,6]。然而，一般而言，DCT 却不如 KLT 有效。

尽管 KLT 在最大化编码增益(或等价地，能量紧缩)方面和信号去相关方面的能力是最佳的，但它仅对已知协方差矩阵的平稳信源才是可以计算的。实际上，信源可以在时间或空间上变化，所以必须不断更新基于前面信号样点的协方差矩阵并重新计算特征矢量，计算上的需求量很大。而且，不存在从任意协方差矩阵中导出 KLT 的快速算法。因此，对于实际应用系统，

---

① 对应于 λ_2 和 λ_3 的特征矢量的解不是惟一的。例如，另一组解是：

$$[\mathbf{\Phi}] = [\phi_1,\phi_2,\phi_3,\phi_4] = \begin{bmatrix} 1/2 & 1/\sqrt{2} & 0 & 1/2 \\ 1/2 & 0 & -1/\sqrt{2} & -1/2 \\ 1/2 & 0 & 1/\sqrt{2} & -1/2 \\ 1/2 & -1/\sqrt{2} & 0 & 1/2 \end{bmatrix}$$

希望采用独立于信号的变换。已经证明对于通常图像信号的协方差矩阵，DCT非常接近KLT。因此，DCT已在图像编码中取代了KLT。

9.1.7 基于DCT的图像编码器和JPEG标准

前面几节已经介绍了变换编码的基本原理。如图9.1所示，变换编码器由三部分组成：变换、量化和二进制编码。到目前为止的分析中，我们假设每个系数是用最佳的概率密度函数标量量化器量化的，且量化的索引是用固定长编码器编码的。但实际上，已发现均匀量化和可变长编码的联合更为有效。这里我们描述一种典型的块变换编码器，它是称为JPEG[1] 的图像编码标准[15,29]和将在第13章介绍的所有视频编码标准的基础。在JPEG标准中，这种方法直接应用于每个图像块，而在视频编码标准中，它应用于当前视频帧的原始图像块与基于前一个视频帧预测的块之间的差值。有时候，当预测不正确或可望重置预测环时它也直接应用于原始图像块。

给定一幅图像，首先把它分成8×8的非重叠图像块，然后对每个块应用点DCT。接着用均匀量化器量化系数，每个量化器具有不同的步长，在一个量化表中规定了不同系数的步长。量化表是基于对不同频率分量的视觉敏感度设计的：一般，低频系数比高频系数分配较短的步长，因为人眼对于低频分量的变化更敏感，如第2章所述。所用的这个特殊的表是在压缩的比特流的开始作为附带信息指定的。通常对亮度和色度分量用不同的表。一个在JPEG标准中给出的亮度量化表的例子如图9.7所示。代替如所给出的那样使用指定的表，我们可以缩放这个表来增加或减小步长，以便达到期望的比特率。缩放因子在JPEG编码器中称为质量因子，而在MPEG和H系列的视频编码器中则称为量化参数(QP)。

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

图9.7 一个亮度分量量化表的例子[15,29]

对于量化的DCT系数的二进制编码，DCT系数按照图9.8所示的"Z"形顺序被排列成一维数组。这种扫描次序把低频系数放在高频系数前面。因为在一般的图像块中许多量化系数是零，所以一个个单独地确定系数值是效率不高的。更合适的是采用游程表示，它以DC值开始，后跟一系列符号[2]。每个符号由两个数组成：从最近的一个非零值算起零的个数(称为游程)和接下来的一个非零值。一个特殊的符号"EOB"放在块的最后一个非零值之后，用来指示

① JPEG标准是指由国际标准组织(ISO)下属的联合图像专家组(JPEG)建议的静止图像压缩的国际标准。

② 如将在第13章中所描述的，在视频编码中所用的游程编码方法稍微有所不同。

块的结束。图 9.9 示出了如何将一个 8×8 块的 DCT 系数转化成这样的符号的一个例子。理想情况下，DC 值和这些符号用两个分离的 VLC 表独立地进行编码。为了避免使用很大的 VLC 表，系数值的动态范围被分割为若干段。非零系数的幅度是由段数和在该段中的相对值确定的。对于 DC 系数，段数是基于不同段的频数被哈夫曼编码的，而相对幅度是用固定长的码字编码的。对于 AC 系数，每个符号被进一步分成两部分：由具有非零值段数的零游程组成的部分用哈夫曼编码，而非零值的相对幅度用固定长编码。VLC 表通常是基于训练数据预先设计的。为了改善编码效率，可以使用算术编码代替哈夫曼编码。

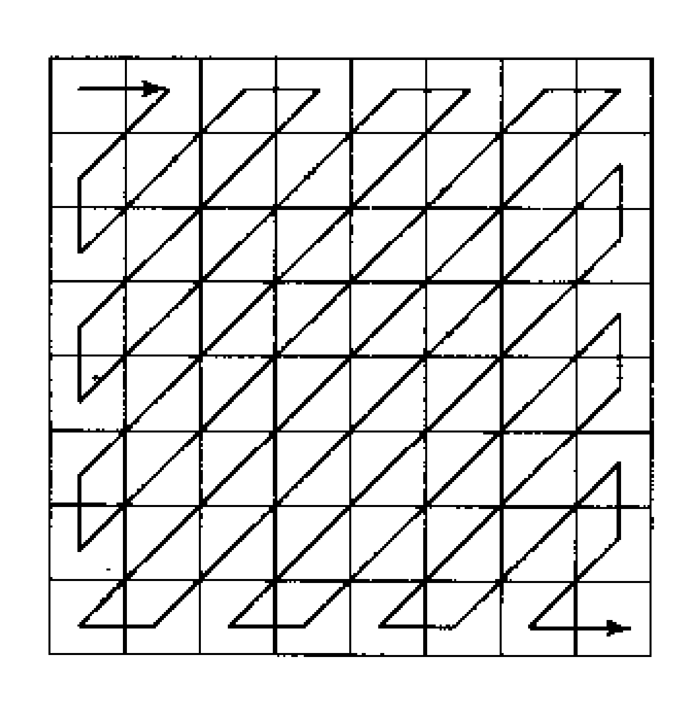


图 9.8　DCT 系数的"Z"形顺序

DCT系数：　[5 0 0 2 3 0 0 4 0 0 0 0 0 0 1 0 0 0 0 0 .. 0]

编码符号：　5, (2,2), (0,3), (2,4), (6,1), EOB

图 9.9　把由"Z"形扫描得到的 64 个量化的 DCT 系数转换成二进制编码符号

当上述方法直接用于原始图像块时，相邻块的 DC 值通常是相似的。为了进一步提高编码效率，一个块的 DC 系数(即平均值)可由前一个编码块的 DC 值进行预测，然后用刚才介绍的方法对 DC 预测误差进行量化和编码。这样，图 9.9 中的值"5"实际上表示 DC 预测误差。同样，图 9.7 中所给出的表的左上角的步长实际上是用于 DC 预测误差的。对于彩色图像，每个彩色分量可以用这种方法分别进行编码。图 9.10 示出了几个由选择不同质量因子得到的 JPEG 编码的图像。这些图像是对图 9.7 给出的量化表进行缩放并使用标准推荐的哈夫曼表得到的。

9.1.8　矢量变换编码

对于到目前为止所讨论的变换编码，我们都假设对变换系数进行标量量化。尽管可以对变换系数应用矢量量化，但没有比 SQ 明显的增益，因为变换系数已经去相关了。进一步提高变换编码增益的方法是用所谓的矢量变换[22]。这种变换把一组矢量样点转换成一组矢量系数，然后对每个系数进行矢量量化。为了使编码效率最大，矢量变换应该在一个矢量系数的变量内引入相关性，但去除相继的矢量系数的相关性。关于图像和视频编码中矢量变换及其应用的更多的信息，读者可以参考 Li 和 Zhang 的综述文章[23]。



图 9.10 用JPEG算法在不同的比特率下编码的图像。左上图是8 bpp 的原始图像;右上图是在 0.59 bpp 下编码的;左下图是在 0.37 bpp 下编码的;而右下图是在 0.22 bpp 下编码的

9.2 预测编码

除了变换编码以外,预测编码是另外一种重要的图像和视频编码方法。事实上,采用运动补偿预测的时间预测编码是现代视频编码标准成功的关键。我们以预测编码的综述开始本节(9.2.1 节)。然后描述如何设计预测器使预测误差最小化,并导出标量量化的预测编码增益(9.2.2 节)。最后两小节阐述如何在图像和视频编码中应用空间和时间预测(9.2.3 节和 9.2.4 节)。

9.2.1 概述

在预测编码中,不是对一个像素直接编码,而是由同一帧或前一帧中的相邻像素值来预测它的值。这是受相邻像素通常有类似的彩色值这一实事启发,因此脱离过去而独立地确定当前值是比特的浪费。图 9.11 示出了一个一般的有损预测编码系统的编码器和解码器方框图。在编码器中,首先由存储在存储器中的几个先前重建的样点来预测输入样点,然后量化预测误差,再用可变长编码器编码。解码器的重建值是预测值加上量化的误差。为了保证编码器和

解码器使用完全相同的预测值，编码器必须重复与解码器再生重建样点同样的过程，这称为闭环预测。这种类型的编码器通常称为差分脉冲编码调制(DPCM)[①]。

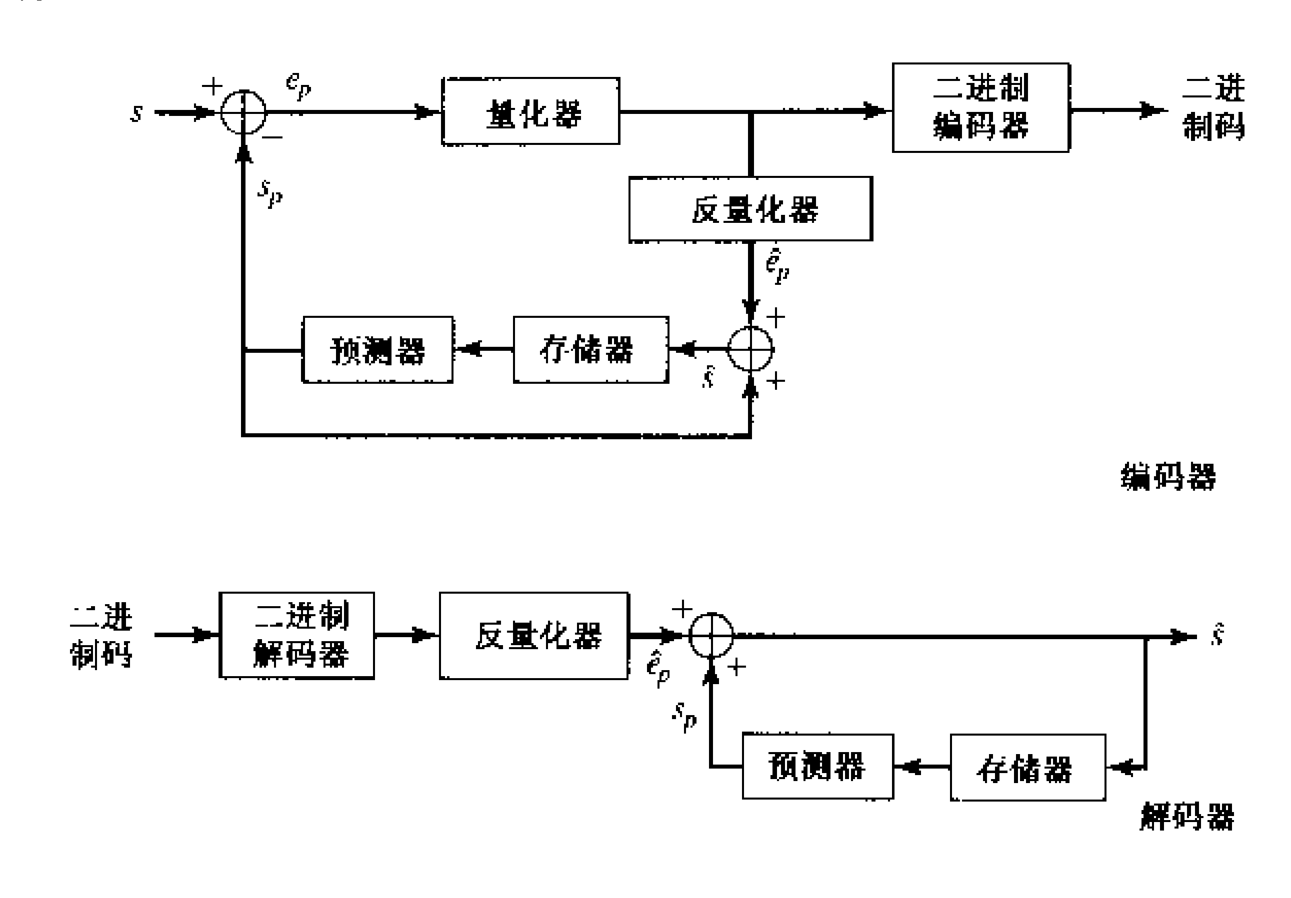


图 9.11　一个有损预测编码系统的编码器和解码器方框图

当预测误差值不经量化而直接编码时，该系统就简化为无损预测编码，这对于要求无失真地存储或传输原始信号的应用系统是非常有用的。

有损预测编码器的误差分析　参考图 9.11。令 s 和 s_p 表示原始样点值和预测样点值，而 $e_p = s - s_p$ 和 $\hat{e}_p$ 表示原始预测误差和量化后的预测误差。令 e_q 表示 e_p 的量化误差，则 $e_p = \hat{e}_p + e_q$。s 的重建值是：

$$\hat{s} = s_p + \hat{e}_p = s_p + e_p - e_q = s - e_q \tag{9.2.1}$$

因此，原始样点值与重建样点值之间的误差是 $e_s = s - \hat{s} = e_q$，与预测误差的量化误差完全一样。于是对于固定的预测器，一个有损预测编码器中的失真完全取决于预测误差的量化器。

为设计误差量化器，可以采用第 8 章中讨论的各种量化技术。因为误差通常具有非均匀分布，所以通常使用关于误差信号分布(通常模型为零均值拉普拉斯分布)优化的非均匀量化器。为改进编码效率，也可以对误差信号采用矢量量化器或变换编码器。

影响预测编码器效率的关键是所用的预测器。由 8.5 节可知，在给定比特率情况下由量化器引入的失真与信号的方差成正比(见公式(8.5.18))。为了使编码效率最大，预测器应该使预测误差的方差为最小。如何设计预测器以达到此目的是下一小节的主题。

9.2.2　最佳预测器设计和预测编码增益

一般地，可以使用各种各样的预测器，线性的或非线性的。但在实际中，为了易于实现几乎无例外地采用线性预测器。令 s_0 表示当前像素，而 $s_k (k = 1, 2, \cdots, K)$ 表示用来预测 s_0 的前

① 传统上，预测误差被均匀量化，然后用称为脉冲编码调制的幅度调制方法传输。随着数字编码和传输技术的出现，预测误差可以非均匀地量化，可以使用各种可变长编码方法，并且可以用不同数字调制技术来传输二进制比特。因此 DPCM 这个名称已不再是这种编码方案的一个准确的描述。

面的像素。在线性预测器中，s_0 的预测值描述为：

$$s_p = \sum_{k=1}^{K} a_k s_k \tag{9.2.2}$$

其中，a_k 称为预测系数。用于预测的样点数 K 通常称为预测器的阶。设计预测编码器的关键问题是如何确定这些系数。如前所述，为了使编码效率最大，应该使预测误差的方差(等价于预测器的 MSE)最小。设 S_k 表示对应 s_k 的随机变量，而 S_p 表示对应 s_p 的随机变量，这个误差定义为：

$$\sigma_p^2 = E\{|\mathcal{S}_0 - \mathcal{S}_p|^2\} = E\left\{\left|\mathcal{S}_0 - \sum_{k=1}^{K} a_k \mathcal{S}_k\right|^2\right\} \tag{9.2.3}$$

最佳预测系数应使这个误差最小。这可以通过令$\partial\sigma_p^2/\partial a_l = 0$实现，由此得到：

$$E\left\{\left(\mathcal{S}_0 - \sum_{k=1}^{K} a_k \mathcal{S}_k\right)\mathcal{S}_l\right\} = 0, \quad l = 1,2,\cdots,K \tag{9.2.4}$$

注意，这个公式正好是线性最小均方误差(LMMSE)估计器的正交性原理。它指出，预测误差必须与每个用来预测的过去的样点正交。令 $R(k,l) = E\{\mathcal{S}_k \mathcal{S}_l\}$ 表示$\mathcal{S}_k$ 与$\mathcal{S}_l$ 之间的相关性。由公式(9.2.4)可以容易导出如下一组线性方程：

$$\sum_{k=1}^{K} a_k R(k,l) = R(0,l), \quad l = 1,2,\cdots,K \tag{9.2.5}$$

或用矩阵形式，

$$\begin{bmatrix} R(1,1) & R(2,1) & \cdots & R(K,1) \\ R(1,2) & R(2,2) & \cdots & R(K,2) \\ \cdots & \cdots & \cdots & \cdots \\ R(1,K) & R(2,K) & \cdots & R(K,K) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdots \\ a_K \end{bmatrix} = \begin{bmatrix} R(0,1) \\ R(0,2) \\ \cdots \\ R(0,K) \end{bmatrix} \tag{9.2.6}$$

或

$$[\mathbf{R}]\mathbf{a} = \mathbf{r} \tag{9.2.7}$$

这个求解 LMMSE 预测器的方程通常称为尤里－沃克(Yule-Walker)方程。

由公式(9.2.7)，最佳预测系数是：

$$\mathbf{a} = [\mathbf{R}]^{-1}\mathbf{r} \tag{9.2.8}$$

这个预测器的 MSE 是：

$$\sigma_p^2 = E\{(\mathcal{S}_0 - \mathcal{S}_p)\mathcal{S}_0\} = R(0,0) - \sum_{k=0}^{K} a_k R(k,0) \tag{9.2.9}$$

$$= R(0,0) - \mathbf{r}^T\mathbf{a} = R(0,0) - \mathbf{r}^T R^{-1}\mathbf{r} \tag{9.2.10}$$

第一个等式是由公式(9.2.4)的正交性原理得到的。

对于平稳信源，像素的自相关是常数，与它的空间位置无关；也就是说，$R(l,l) = R(0,0)$，$l = 1,2,\cdots,K$。而且，两个像素之间的相关性是对称的；即 $R(k,l) = R(l,k)$。矩阵 R 实际上是预测矢量$\boldsymbol{\mathcal{S}}_p = [\mathcal{S}_1, \mathcal{S}_2, \cdots, \mathcal{S}_K]^T$ 的自相关矩阵，而 $\mathbf{r} = E\{\mathcal{S}_0 \boldsymbol{\mathcal{S}}\}$是当前像素$\mathcal{S}_0$ 与$\boldsymbol{\mathcal{S}}_p$ 之间的相关性。当信源是一维平稳信源时，$\mathcal{S}_0 = \mathcal{S}(n)$是信源的一个任意样点，而$\mathcal{S}_k = \mathcal{S}(n-k)$是$\mathcal{S}(n)$前面的第 k 个样点，我们有 $R(k,l) = E\{\mathcal{S}(n-k)\mathcal{S}(n-l)\} = R_s(k-l)$，它是距离 $k-l$ 的两个样点之间的相关。在这种情况下，矩阵$[\mathbf{R}]$是 Toeplitz 的，同一个对角线

上的所有元素都相等，并且相应于上三角和下三角的线是相互共轭的。这样的矩阵可以用莱文森－德宾(Levinson-Durbin)算法进行有效地求逆[30]。如果信源是二维的，矩阵[**R**]一般不会有如此好的结构。但一般仅用有限数目的过去样点进行预测，因为两个像素之间的相关随它们距离的增加而迅速减小。通常，最多用 4 个像素（当前像素的左、上、左上、右上）进行预测。在这种情况下，直接求解上述方程是一件很容易的事情。

最佳预测器的这个解假设预测值 s_p 是由原始的过去样点 s_k 得到的。在有损预测编码器中，必须用重建的过去样点 $\hat{s}_k$ 代换 s_k。理想情况下，公式(9.2.3)中的误差应该被替换为：

$$\sigma_p^2 = E\left\{\left|\mathcal{S}_0 - \sum_k a_k \hat{\mathcal{S}}_k\right|^2\right\} \tag{9.2.11}$$

但因为 $\hat{\mathcal{S}}_k$ 是以一种复杂的关系依赖于预测器和量化器，所以求使这个误差为最小的解是很难的。实际上，我们简单地假设量化误差是可以忽略的，并且通过使公式(9.2.3)中的误差为最小来设计预测器。

预测编码增益　与变换编码一样，我们可以把预测编码对 PCM 的增益定义为在相同比特率下由这两种编码器所产生的失真比。回忆对于预测编码，被量化和编码的是预测误差，且原始信号的失真等于预测误差的量化误差。假设预测误差采用最佳标量量化，失真与码率的关系为：

$$D_{\mathrm{DPCM}} = \epsilon_p^2 \sigma_p^2 2^{-2R} \tag{9.2.12}$$

其中 ϵ_p^2 取决于预测误差的概率密度函数，且 σ_p^2 是预测误差的方差，等于预测器的 MSE。另一方面，PCM 的率失真函数由公式(9.1.45)给出。因此，预测编码增益是：

$$G_{\mathrm{DPCM}} = \frac{D_{\mathrm{PCM}}}{D_{\mathrm{DPCM}}} = \frac{\epsilon_s^2 \sigma_s^2}{\epsilon_p^2 \sigma_p^2} \tag{9.2.13}$$

一般地，预测误差 σ_p^2 随预测器阶数 K 的增加而减小。

对于由公式(9.2.8)给出的 LMMSE 预测器，可以证明[19]，当预测器的阶趋于无穷时，最小预测误差与信号 $S(e^{j\omega})$ 的功率谱密度(PSD)① 的关系为：

$$\sigma_{p,\min}^2 = \lim_{K\to\infty} \sigma_p^2 = \exp\left(\frac{1}{2\pi}\int_{-\pi}^{\pi} \log_e S(e^{j\omega})\,d\omega\right) \tag{9.2.14}$$

比值 $\gamma_s = \sigma_{p,\min}^2 / \sigma_s^2$ 称为信号的谱平坦度测度。因此，预测编码增益与谱平坦度成反比。这是因为具有集中的窄带谱的信号更易于预测。具有平坦谱的信号（即白噪声）是不可预测的。

可以证明公式(9.2.14)中的积分可以写为：

$$\sigma_{p,\min}^2 = \lim_{K\to\infty}\left(\prod_k \lambda_k\right)^{1/N} \tag{9.2.15}$$

其中，λ_k 是信号的 N 阶协方差矩阵的第 k 个特征值②。另一方面，信号方差等于：

$$\sigma_s^2 = \lim_{K\to\infty} \frac{1}{K}\sum_k \lambda_k \tag{9.2.16}$$

因此，当 $K\to\infty$ 时的预测编码增益可以表达为：

---

① 平稳随机序列 $\{\mathcal{S}(n)\}$ 的 PSD 是它的自相关函数 $R(k) = E\{\mathcal{S}(n)\mathcal{S}(n-k)\}$ 的傅里叶变换。

② 可以证明 $\lambda_k = S(e^{j\omega_k})$，其中 ω_k 通常是区间 $(-\pi,\pi)$ 上的非等距样点。

$$\lim_{K\to\infty} G_{\mathrm{DPCM}} = \frac{\epsilon_s^2}{\epsilon_p^2} \frac{\lim_{K\to\infty} \frac{1}{K}\sum_k \lambda_k}{\lim_{K\to\infty}\left(\prod_k \lambda_k\right)^{\frac{1}{K}}} \tag{9.2.17}$$

它等于当变换长度 N 趋于无穷时用 KLT 可以达到的变换编码增益(见公式(9.1.47))。因此,当变换长度和预测器阶数都趋于无穷时,这两种方法等价。然而,对于任何有限的 N,长度为 N 的变换编码比 N 阶预测编码的效率低[19]。这是因为长度为 N 的变换一次仅涉及 N 个样点,而因为闭环预测中量化误差的反馈,任意阶的预测编码器都具有无限的有效记忆。

9.2.3 空间域线性预测

令$\psi(m,n)$表示一个视频帧。在空间域的线性预测中,当前像素$\psi(m,n)$是由它的过去的相邻像素$\psi(k,l),(k,l)\in \mathcal{B}_{(m,n)}$预测的,其中,$\mathcal{B}_{(m,n)}$表示像素$(m,n)$的因果性邻域。如果$\psi(m,n)$之间的相关性和$\mathcal{B}_{(m,n)}$中的每个像素都已知,可以根据公式(9.2.6)确定最佳预测器。

例 9.6 再一次考虑图 9.6 所示的图像信源。这一次用预测编码器,它由 $s_1=\psi(m,n-1)$(像素 C),$s_2=\psi(m-1,n)$(像素 B)和 $s_3=\psi(m-1,n-1)$(像素 A)来预测像素 $s_0=\psi(m,n)$(像素 D)。在这种情况下,公式(9.2.6)简化为:

$$\begin{bmatrix} R(C,C) & R(C,B) & R(C,A) \\ R(B,C) & R(B,B) & R(B,A) \\ R(A,C) & R(A,B) & R(A,A) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} R(D,C) \\ R(D,B) \\ R(D,A) \end{bmatrix}$$

或

$$\begin{bmatrix} 1 & \rho_d & \rho_v \\ \rho_d & 1 & \rho_h \\ \rho_v & \rho_h & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \rho_h \\ \rho_v \\ \rho_d \end{bmatrix}$$

在 $\rho_h=\rho_v=\rho,\rho_d=\rho^2$ 的特殊情况下,最佳预测器是:

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \rho \\ \rho \\ -\rho^2 \end{bmatrix}$$

应用公式(9.2.10),这个预测器的 MSE 是:

$$\sigma_p^2 = R(0,0) - [R(0,1) \quad R(0,2) \quad R(0,3)] \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = (1-\rho^2)^2\sigma_s^2$$

对于 $R=2$ 比特的最佳标量量化,量化误差是:

$$D = \epsilon_{\mathrm{Gaussian}}^2 \sigma_p^2 2^{-2R} = 0.000594\, \epsilon_{\mathrm{Gaussian}}^2 \tag{9.2.18}$$

相对于 PCM 的编码增益是:

$$G_{\mathrm{DPCM}} = \frac{\sigma_s^2}{\sigma_p^2} = \frac{1}{(1-\rho^2)^2} = 105.19 \tag{9.2.19}$$

把这个结果与例 9.4 和例 9.5 中的结果相比较,我们看到在这三种编码时都涉及 2×2 像

素块的方法中,预测编码方法的效率最高。这与理论结果是一致的,即采用 N 阶预测器的预测编码好于采用长度为 N 的变换的变换编码。注意,理论分析和例子都假设用最佳的标量量化,这在实际中可能是不能实现的。预测编码的主要问题是,编码比特流中影响单个样点的传输差错可能导致编码器与解码器之间的失配,以至于所有后面的重建样点可能都是错的。这称为误码扩散效应。传输误码对编码比特流的影响以及防止或抑制这种效应的机制将在第14章中讨论。

注意,空间预测不仅可以应用于原始像素值,也可以应用于变换系数。例如,在块变换编码器中,相邻块的DC系数(即均值)时常是相似的,我们可以由当前块的上方和左侧的块的DC值来预测当前块的DC值。在9.1.7节讨论的JPEG图像编码标准中,是用左侧块的DC值预测的。通常,相邻块的AC系数之间的相关性不足以保证预测之用。

在采用小波变换(第11章)的编码器中,相同空间位置但不同尺度的系数时常是相关的。因此,可以在这些系数之间进行预测。在那里,经常使用一种非线性预测器,不是预测实际的系数值,而是由较粗尺度中非零系数的位置预测较细尺度中非零系数的位置。

9.2.4 运动补偿时间预测

单向时间预测 除了在帧内应用预测外,当前帧的像素值也可以由它所对应的前一帧的像素进行预测。令 $\psi(\mathbf{x},t)$ 表示第 t 帧在像素 $\mathbf{x}$ 的像素值,并且令 t_- 表示前一帧的时刻。则预测过程描述为:

$$\psi_p(\mathbf{x},t)=\psi(\mathbf{x},t_-) \tag{9.2.20}$$

这称为线性时间预测。仅当场景静止时,这样的预测才是有效的。在一个真实世界的视频中,场景中的物体以及摄像机通常都是运动的,使得两个相邻帧中相同空间位置的像素可能有非常不同的值。在这种情况下,运动补偿预测(MCP,motion-compensated prediction)更适用,它用下式预测:

$$\psi_p(\mathbf{x},t)=\psi(\mathbf{x}+\mathbf{d}(\mathbf{x}),t_-) \tag{9.2.21}$$

其中,$\mathbf{d}(\mathbf{x})$ 表示像素 $\mathbf{x}$ 从时刻 t 到 t_- 的运动矢量。用第6章介绍的术语,帧 t 是锚定帧,帧 t_- 是目标帧。在视频编码著作中,帧 t_- 通常称为参考帧,帧 t 称为编码帧或当前帧,而 $\psi_p(\mathbf{x},t)$ 称为预测帧。参考帧必须在编码帧之前被编码并重建。

在使用MCP的编码器中,必须确定MV和预测误差。很明显,如果在每个像素上确定MV和预测误差值,就不得不使用比确定每个像素的原始像素值更多的比特。在实际编码器中,用更紧凑的运动表示来降低运动编码的比特率。一种流行的视频编码方法是采用基于块的运动表示,而每个块仅编码一个MV(我们将在9.3节中讨论基于块的视频编码器)。

到目前为止所介绍的时间预测器仅使用前一帧的一个像素。这个限制主要是由于实际实现的约束造成的,存储多于一个的前面的帧经常成本太高。然而,理论上来讲,使用多个前面的帧的像素可以提高预测精度。在这种情况下,仍可以用公式(9.2.6)中所给出的最佳预测器设计解决方案,以确定与前面各个像素有关的预测器系数。

双向时间预测 事实上,不需要预测器只依赖于前面的帧。在双向时间预测中,当前帧的像素是由前一帧 t_- 的一个像素以及后一帧 t_+ 的一个像素预测的。帧 t 的预测值描述为:

$$\psi_p(\mathbf{x},t)=a_-\psi(\mathbf{x}+\mathbf{d}^-(\mathbf{x}),t_-)+a_+\psi(\mathbf{x}+\mathbf{d}^+(\mathbf{x}),t_+) \tag{9.2.22}$$

其中,$\mathbf{d}^-(\mathbf{x})$ 和 $\mathbf{d}^+(\mathbf{x})$ 分别表示 $\mathbf{x}$ 从 t 到 t_- 和从 t 到 t_+ 的运动矢量(MV)。一般,我们称由前

面的($t_- < t$)参考帧预测当前帧为前向运动补偿,而称由未来的($t_+ > t$)参考帧预测当前帧为后向运动补偿[①]。理想情况下,应该用公式(9.2.6)所给出的最佳预测器解来确定系数 a_- 和 a_+,这要求已知在所涉及的三帧中对应像素之间的相关性。实际上,它们通常是试探地选择的。为了使预测值具有与原始值一样的均值,它们被选择为使 $a_- + a_+ = 1$。

当编码帧中有被物体运动露出的区域时,采用双向时间预测可能是非常有益的。这样的区域在前一帧没有对应的区域,只在后一帧有对应的区域。例如,如果回过头去看图 5.12,可以看出在帧 $k+1$ 露出的背景在帧 k 中没有对应的背景,但将在帧 $k+2$ 中有对应的背景。预测系数 a_- 和 a_+ 通常是基于预测精度调节的。在刚才考虑的例子中,可以选择 $a_- = 0$, $a_+ = 1$。当在帧 t_- 和帧 t_+ 中都可以找到对应时,一个好的选择是 $a_- = a_+ = 0.5$。与单向预测一样,可以在块级进行双向预测,所以对于一个块只须确定两个 MV。

注意,使用双向预测需要用不同于原时间顺序的一个顺序来进行帧的编码。例如,可以首先用单向预测仅用过去的编码帧来编码一个帧的子集,然后再用双向预测编码其余的帧。这种类型的脱离顺序编码如图 9.12 所示。尽管双向预测可以提高预测精度和编码效率,但它会引入编码延迟,所以一般在实时应用系统中(例如视频电话或视频会议)是不用的。例如,用于交互式通信的 H.261/H.263 标准仅用单向预测和有限制的双向预测(称为 PB 模式)。然而主要用于视频发布的 MPEG 标准系列既采用单向预测也采用双向预测。

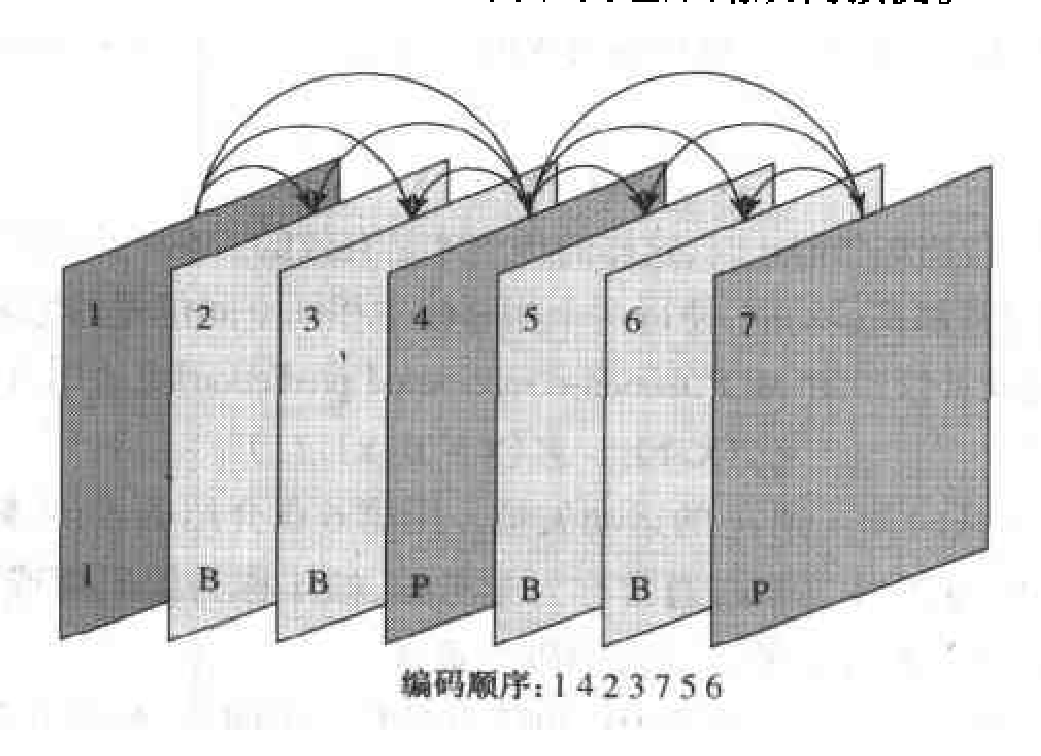


图 9.12 采用单向和双向时间预测的视频编码。箭头指示用于预测编码帧的参考帧。标号为 I,P 和 B 的帧分别是不用预测、用单向预测和用双向预测编码的

9.3 采用时间预测和变换编码的视频编码

一种流行的和有效的视频编码方法是用基于块的时间预测和变换编码。这个方法实际上是所有国际视频编码标准的核心。在本节中,我们首先给出这种编码方法的综述(9.3.1 节)。紧跟着是对它的一个扩展,使用更复杂的运动补偿预测(9.3.2节)。接下来我们考虑在由应

① 这些不应该与 6.2 节中所定义的后向和前向运动估计相混淆。

用所提出的码率约束下如何选择这种编码方法中的各种编码参数,以便使编码效率最大化的问题(9.3.3 及 9.3.4 节)。作为可选的先进课题,我们分析运动估计精度和环形滤波对编码效率的影响(9.3.5 节)。

9.3.1 基于块的混合视频编码

在这种编码器中,每个视频帧被分成固定大小的块,且每个块或多或少地独立地进行处理,因此称为"基于块的"。"混合"这个词意味着每个块是联合运用运动补偿时间预测和变换编码进行编码的。图 9.13 给出了这种编码范例中的关键步骤。首先,用基于块的运动估计由前面已编码的参考帧对块进行预测。运动矢量确定当前块和最佳匹配块之间的位移。预测块是由前面的帧基于估计的 MV 用运动补偿[①]获得的。然后,用 DCT 对预测误差块进行变换[②],量化 DCT 系数,并用可变长编码把它们转化成二进制码字。实际的编码方法在原则上类似于 9.1.7 节中所描述的。与 JPEG 标准一样,DCT 系数的量化是由一个量化参数控制的,它对预先定义的量化表进行缩放。各种视频编码标准所使用的量化表、量化器实现和游程编码方法将在第 13 章描述。注意,编码器必须仿效解码器的操作以导出与解码器一样的重建帧,从而在用于预测的参考帧之间不会失配。

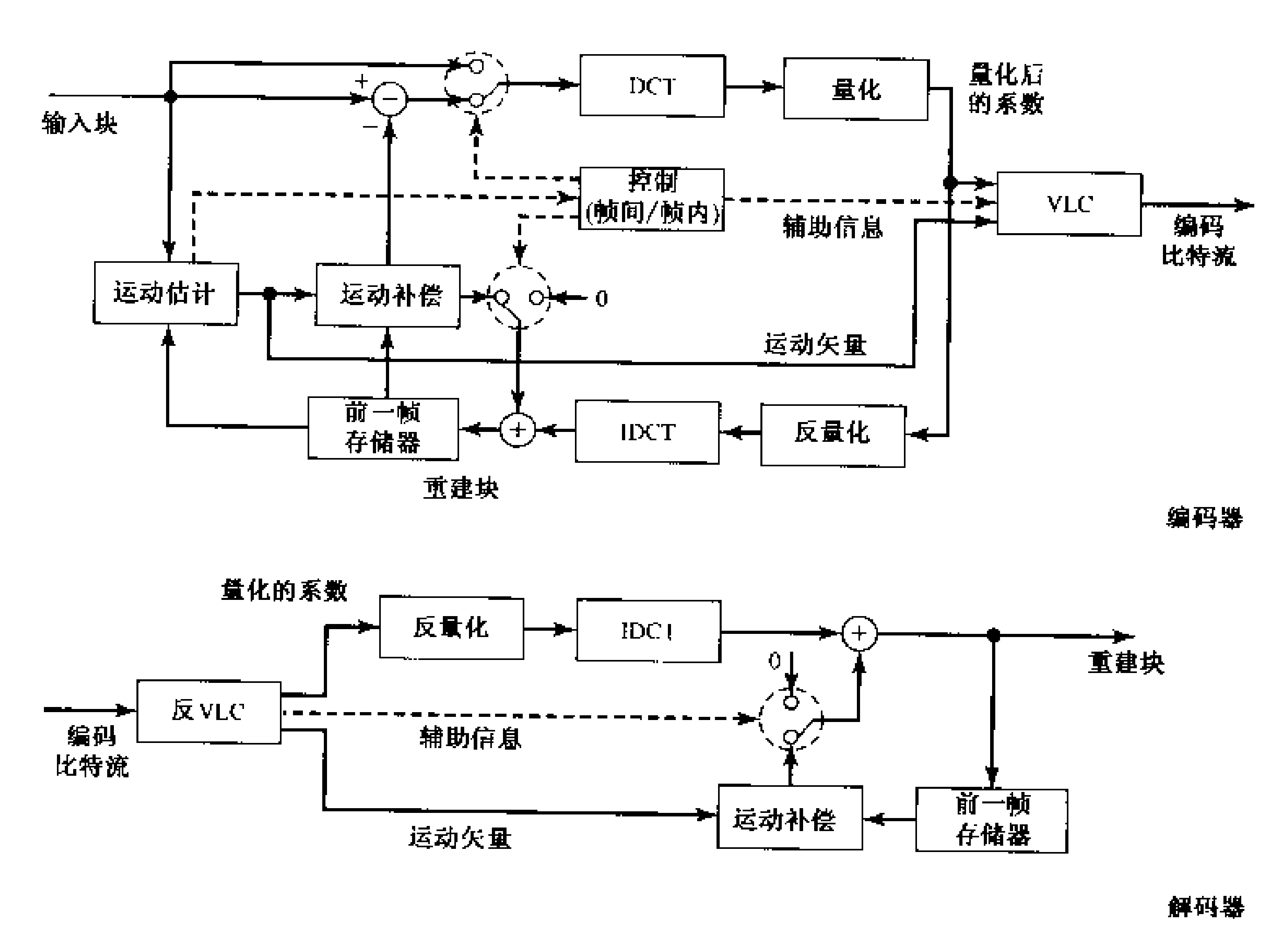


图 9.13　在一个典型的基于块的混合编码系统中块的编码和解码过程

① 短语"运动补偿"在此处以及本书的后面部分中都指的是运动补偿预测。

② 理想情况下,如果预测非常准确,那么误差模式将接近白噪声,可以用标量量化有效地进行编码。然而实际上,基于块的运动模型的使用经常导致结构化的误差模式。使用变换编码是为了利用误差像素之间的相关性。

上述讨论假设时间预测是成功的，预测误差块要求用比原始图像块少的比特进行编码。这种编码方法称为 P 模式。当情况不是如此时，原始块将直接用变换编码进行编码。这称为帧内模式。可以把用单个参考帧预测换成双向预测，它要找两个最佳的匹配块，一个在前面的帧中，另一个在后面的帧中，且用这两个匹配块的加权平均作为当前块的预测。在这种情况下，两个 MV 与每个块都有联系。这称为 B 模式。P 和 B 模式一般称为帧间模式。模式信息和 MV 以及其他的关于图像格式的辅助信息、块位置等也用 VLC 编码。

实际上，用于运动估计的块大小可能与用于变换编码的不一样。一般，运动估计是在一个较大的块（称为宏块）上进行的，宏块（MB）被进一步分成几个块，对这些块求 DCT 的值。例如，在大多数视频编码标准中，宏块的大小是 16×16 个像素，而每个块的大小是 8×8 个像素。如果彩色亚采样格式是 4:2:0，那么每个宏块由 4 个 Y 块、1 个 Cb 块和 1 个 Cr 块组成。编码模式（即帧内或帧间模式）是在宏块水平上确定的。因为相邻的宏块或块的 MV 和 DC 系数通常是类似的，所以一般用前一个宏块或块的 MV 和 DC 系数作为预测的值来对它们进行预测编码。在第 13 章中描述的所有视频编码标准中，若干个宏块形成一个块组（GOB）或一个片，几个块组或片形成一幅图像。在不同的视频编码标准和图像尺寸中，块组和片的大小和形状是不同的，经常是为应用的需要而特制的。MV 和 DC 系数的预测通常限制在同一个块组或片内。因此，为块组中第一个宏块或块预测的 MV 或 DC 值被置为某个默认值。这是为了当压缩的比特流因传输或存储差错而损坏时抑制误码扩散。

一个帧能够以帧内模式整个地进行编码，这样的帧称为内帧或 I 帧。序列中的第一个帧总是作为内帧进行编码的。在采用高比特率或具有松弛的实时约束的应用系统中，也周期地使用内帧以阻止潜在的误码扩散，并使随机访问成为可能。（低等待时间的应用系统，如视频会议，不能用这样的周期刷新，因为来自内帧的比特数通常比预测帧的大得多，这将引起传输数据的突然巨涌。）P 帧仅用过去的帧进行预测，取决于预测精度，一个宏块既可以用帧内模式也可以用 P 模式进行编码，这通常是基于预测误差决定的。最后，B 帧采用双向预测，而 B 帧中的宏块可以用帧内模式、P 或 B 模式进行编码。B 帧只能在周围的 I 帧或 P 帧编码之后进行编码（见图 9.12）。P 帧和 B 帧二者通常都称为间帧。

在 MPEG-1 和 MPEG-2 标准中，把帧被划分成图像组（GOP），而每个图像组以 I 帧开始，后跟交织的 P 帧和 B 帧（具有代表性的图像组如图 9.12 所示）。这使随机访问成为可能：可以访问任何图像组而不需对前面的图像组解码。图像组结构也允许快进和快倒：仅解码 I 帧或解码 I 和 P 帧就可以实现快进。以向后的顺序仅解码 I 帧就可以实现快倒。

9.3.2 重叠块运动补偿

在本小节和下小节中，为了符号的简化，我们不是指定视频函数中的时间索引而是用 $\psi(\mathbf{x})$，$\psi_r(\mathbf{x})$ 和 $\psi_p(\mathbf{x})$ 分别表示编码帧、参考帧和预测帧。在基于块的运动补偿中，块中的每个像素用参考帧中的一个像素通过下式预测：

$$\psi_p(\mathbf{x}) = \psi_r(\mathbf{x} + \mathbf{d}_m), \quad \mathbf{x} \in \mathcal{B}_m \tag{9.3.1}$$

其中，$\mathbf{d}_m$ 表示块 $\mathcal{B}_m$ 的 MV。当 $\mathbf{d}_m$ 不是整数时，需要内插得到 $\psi_r(\mathbf{x} + \mathbf{d}_m)$。如前所述，当估计的 MV 不准确或当块平移模型不适用时，预测图像将受到方块效应的损害。目前已采用了两种方法来克服这个问题。一个是与邻块的运动估计相结合以产生给定像素的预测值，另一个是对预测图像进行滤波。在本节中讨论第一种方法，在 9.3.5 节中讨论滤波方法。

对于重叠块运动补偿(OBMC)[25,26],一个像素的预测不仅基于它所属块的 MV 估计,而且还基于其相邻块的 MV,如图 9.14 所示。令 K 是所考虑的相邻块的总数,并令$\mathcal{B}_{m,k}$和 $\mathbf{d}_{m,k}$,$k\in\mathcal{K}=\{0,1,2,\cdots,K\}$表示第 k 个邻块和它的 MV,且$\mathcal{B}_{m,0}=\mathcal{B}_m$,$\mathbf{d}_{m,0}=\mathbf{d}_m$。预测值由下式确定:

$$\psi_p(\mathbf{x}) = \sum_{k\in\mathcal{K}} h_k(\mathbf{x})\psi_r(\mathbf{x}+\mathbf{d}_{m,k}),\mathbf{x}\in\mathcal{B}_m \tag{9.3.2}$$

系数 $h_k(\mathbf{x})$可以认为是分配给基于 $\mathbf{d}_{m,k}$的估计值的权重。直觉上,对于给定的 $\mathbf{x}$,$h_k(\mathbf{x})$应该与 $\mathbf{x}$ 到$\mathcal{B}_{m,k}$的中心位置的距离成反比。例如,对于图 9.14 中所画出的邻域,对于 $\mathbf{x}_0$,$\mathbf{d}_{m,1}$和 $\mathbf{d}_{m,4}$的权重应该比 $\mathbf{d}_{m,2}$和 $\mathbf{d}_{m,3}$的大。

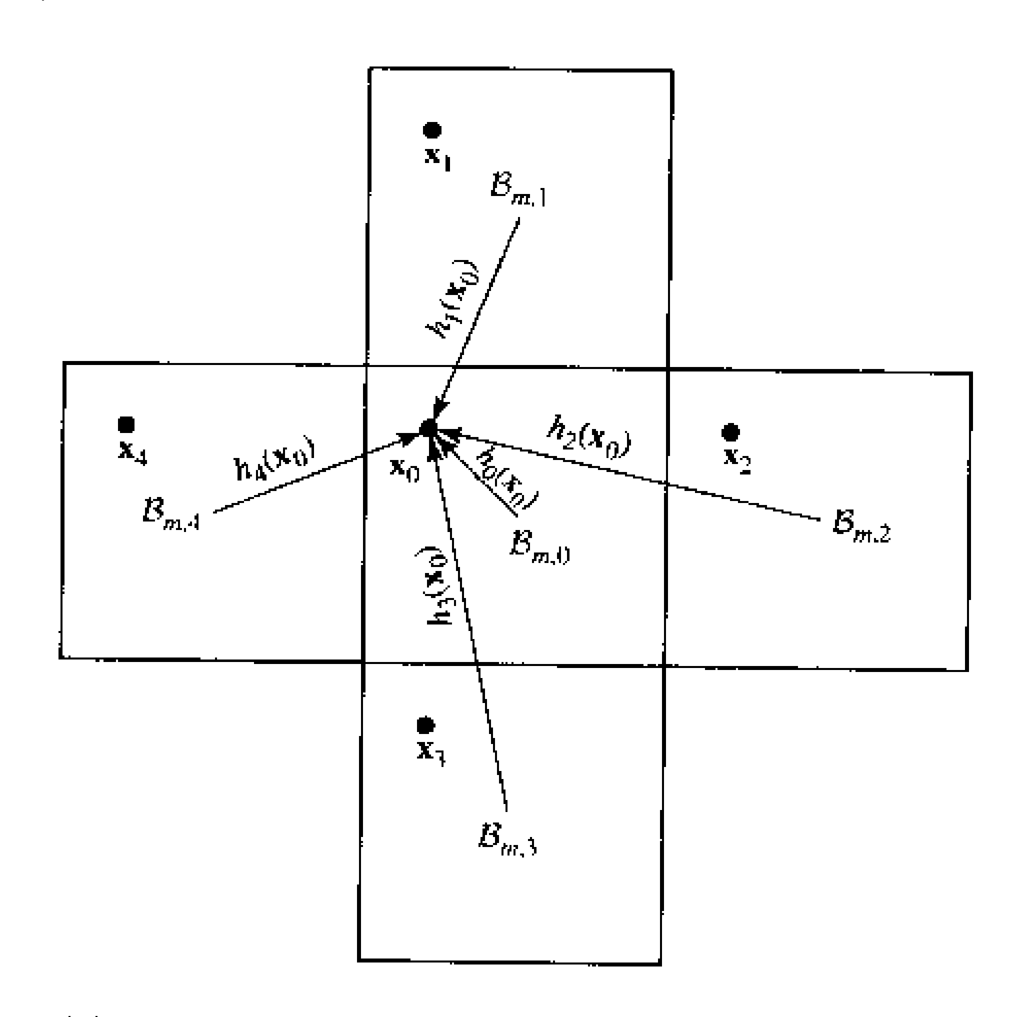


图 9.14　一个用四个邻块进行重叠块运动补偿的例子

从估计理论的观点来看,公式(9.3.2)中的预测器可以解释为给定观测$\psi_r(\mathbf{x}+\mathbf{d}_{m,k})$条件下$\psi(\mathbf{x})$的一个线性估计器。对于给定的 $\mathbf{x}$,最佳估计器 $\mathbf{h}(\mathbf{x})=[h_k(\mathbf{x}),k\in\mathcal{K}]^T$ 可以通过使均方预测误差:

$$E\left\{\left|\psi(\mathbf{x}) - \sum_{k\in\mathcal{K}} h_k(\mathbf{x})\psi_r(\mathbf{x}+\mathbf{d}_{m,k})\right|^2\right\} \tag{9.3.3}$$

为最小来确定。为了保持估计图像的均值,必须约束权重,使得:

$$\sum_{k\in\mathcal{K}} h_k(\mathbf{x}) = 1$$

用拉格朗日乘子法[26]把这个约束合并到优化公式中,得:

$$\mathbf{h}(\mathbf{x}) = [\mathbf{R}(\mathbf{x})]^{-1}\left(\mathbf{r}(\mathbf{x}) - \frac{\mathbf{i}^T[\mathbf{R}(\mathbf{x})]^{-1}\mathbf{r}(\mathbf{x})-1}{\mathbf{i}^T[\mathbf{R}(\mathbf{x})]^{-1}\mathbf{i}}\mathbf{i}\right) \tag{9.3.4}$$

其中,$\mathbf{i}$ 是所有元素等于 1 的 K 维列矢量,$[\mathbf{R}(\mathbf{x})]$和 $\mathbf{r}(\mathbf{x})$分别是自相关矩阵和互相关矢量,其元素为:

$$R_{k,l}(\mathbf{x}) = E\{\psi_r(\mathbf{x}+\mathbf{d}_{m,k})\psi_r(\mathbf{x}+\mathbf{d}_{m,l})\}, k, l \in \mathcal{K}$$
$$r_k(\mathbf{x}) = E\{\psi(\mathbf{x})\psi_r(\mathbf{x}+\mathbf{d}_{m,k})\}, k \in \mathcal{K}$$

可以用这个解为块中的每个像素 $\mathbf{x}$ 导出各个邻块 MV 的权重。上述公式中的数学期望是就图像亮度和运动的不确定性而论的。上述统计量可以由实际视频数据估计。如符号所表示的,它们一般是空间变化的。然而,假设$\psi(\mathbf{x})$是平稳的,这些统计量就仅取决于块中像素的相对位置,不取决于相对于整幅图像的绝对位置。如果像素之间的相关是对称的,那么解 $h_k(\mathbf{x})$ 也是对称的,从而只需要确定左上方 1/4 块中像素的解。

一般情况下,OBMC 可以用于由任何方法估计的块 MV,包括简单的块匹配法(BMA)。然而,如果运动估计是顾及到 OBMC 实现的,则可以进一步减少预测误差。理想情况下,应该使以下修正的 MCP 误差最小化:

$$\sum_{\mathbf{x}\in\mathcal{B}_m}\left|\psi(\mathbf{x}) - \sum_{k\in\mathcal{K}} h_k(\mathbf{x})\psi_r(\mathbf{x}+\mathbf{d}_{m,k})\right|^p \tag{9.3.5}$$

因为块误差取决于它的相邻块的 MV,所以不能像 BMA 那样一次求一个块的 MV。一种方法是迭代地进行运动估计和 OBMC[26]。首先,用传统的 BMA 生成一个 MV 的初始集,这里,预测误差是假设把公式(9.3.1)用于 MCP 而计算的。然后,用这些运动估计进行重叠块运动补偿。在下一次迭代中,对于每个块,通过使公式(9.3.5)中给出的误差为最小来估计 MV,这里,假设相邻块的 MV 与前一次迭代中所得到的一样。

若不用迭代,一种较简单的方法是当用 BMA 计算预测误差时,对当前块以及相邻块的像素加不相等的权重。为了确定适当的权重函数,应该注意当进行 OBMC 时,块$\mathcal{B}_m$ 的运动矢量 $\mathbf{d}_m$ 不仅影响本块,也影响它的相邻块$\mathcal{B}_{m,k}$。因为 $\mathbf{d}_m$ 对$\mathcal{B}_{m,k}$中像素 $\mathbf{x}$ 的贡献被 $h_k(\mathbf{x})$加权,所以我们可以使以下误差最小化:

$$\sum_{k\in\mathcal{K}}\sum_{\mathbf{x}\in\mathcal{B}_{m,k}} |\psi(\mathbf{x}) - \psi_r(\mathbf{x}+\mathbf{d}_m)|^p h_k(\mathbf{x}) \tag{9.3.6}$$

令$\bar{\mathcal{B}}_m = \bigcup_{k\in\mathcal{K}}\mathcal{B}_{m,k}$是包括当前块$\mathcal{B}_{m,0}$及其邻块$\mathcal{B}_{m,k}$, $k \in \mathcal{K}$ 的超块,则这个误差可以重写为:

$$\sum_{\mathbf{x}\in\bar{\mathcal{B}}_m} |\psi(\mathbf{x}) - \psi_r(\mathbf{x}+\mathbf{d}_m)|^p \bar{h}(\mathbf{x}) \tag{9.3.7}$$

其中,窗函数 $\bar{h}(\mathbf{x})$与 $h_k(\mathbf{x})$有关。例如,在图 9.14 中,$\bar{h}(\mathbf{x}_0) = h_0(\mathbf{x}_0)$,$\bar{h}(\mathbf{x}_1) = h_3(\mathbf{x}_0)$,$\bar{h}(\mathbf{x}_2) = h_4(\mathbf{x}_0)$,$\bar{h}(\mathbf{x}_3) = h_1(\mathbf{x}_0)$和 $\bar{h}(\mathbf{x}_4) = h_2(\mathbf{x}_0)$。公式(9.3.7)中的误差是在超块$\bar{\mathcal{B}}_m$上的一个加权的 MCP 误差,且窗函数 $\bar{h}(\mathbf{x})$是原始权重系数 $h_k(\mathbf{x})$的一个重新排列。由 Orchard 和 Sullivan [26]确定的最佳的窗函数示于图 9.15。

试验结果表明当 OBMC 与标准 BMA 相结合时,用 OBMC 可以提高预测图像质量达1 dB。当应用迭代运动估计时,可以取得 2 dB 的改善[26]。由于显著地改善预测精度,OBMC 已作为一个先进选项加入 ITU-T H.263 视频编码标准中。该标准使用图 9.14 所示的四邻域结构,但对每个给定的像素仅两个邻域具有非零权重。用于该标准的为便于快速计算[17]而选择的权重系数 $h_k(\mathbf{x})$示于图 9.16。等价的窗函数 $\bar{h}(\mathbf{x})$如图 9.17 所示。

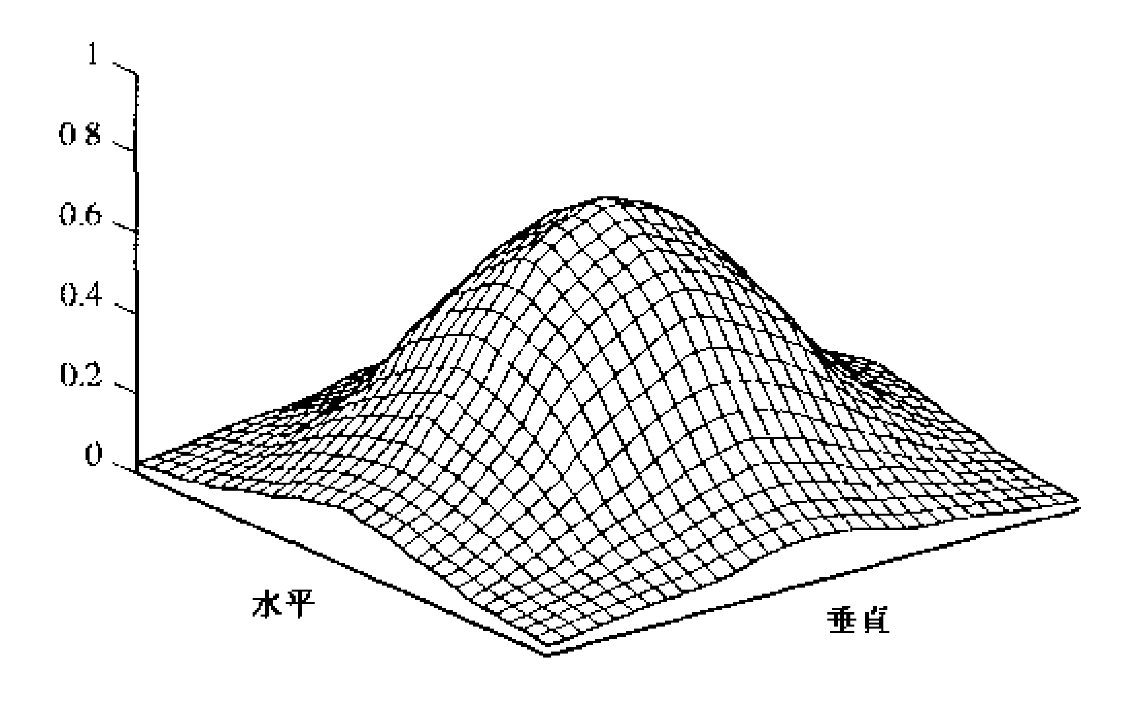


图 9.15　由 Orchard 和 Sullivan 确定的 OBMC 的最佳窗函数

(a)

4	5	5	5	5	5	5	4
5	5	5	5	5	5	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	6	6	6	6	5	5
5	5	5	5	5	5	5	5
4	5	5	5	5	5	5	4

(b)

2	2	2	2	2	2	2	2
1	1	2	2	2	2	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	2	2	2	2	1	1
2	2	2	2	2	2	2	2

(c)

2	1	1	1	1	1	1	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	2	1	1	1	1	2	2
2	1	1	1	1	1	1	2

图 9.16　在 H.263 视频编码标准中[17]规定的 OBMC 的权重系数：(a)用于由当前块运动矢量进行预测；(b)用于由当前块的上方或下方块的运动矢量进行预测；(c)用于由当前块的左边或右边块的运动矢量进行预测。给出的数字是 8× 实际权重。例如，要预测块的左上角的像素，与当前 MB、上方 MB、左边 MB 的 MV 相关联的权重分别是 4/8，2/8 和 2/8。对于位于第一行和第二列的像素，权重分别是 5/8，2/8，1/8

				1	1	1	1	1	1	1	1				
				1	1	1	1	1	1	1	1				
				1	1	2	2	2	2	1	1				
				2	2	2	2	2	2	2	2				
1	1	1	2	4	5	5	5	5	5	5	4	2	1	1	1
1	1	2	2	5	5	5	5	5	5	5	5	2	2	1	1
1	1	2	2	5	5	6	6	6	6	5	5	2	2	1	1
1	1	2	2	5	5	6	6	6	6	5	5	2	2	1	1
1	1	2	2	5	5	6	6	6	6	5	5	2	2	1	1
1	1	2	2	5	5	6	6	6	6	5	5	2	2	1	1
1	1	2	2	5	5	5	5	5	5	5	5	2	2	1	1
1	1	1	2	4	5	5	5	5	5	5	4	2	1	1	1
				2	2	2	2	2	2	2	2				
				1	1	2	2	2	2	1	1				
				1	1	1	1	1	1	1	1				
				1	1	1	1	1	1	1	1				

图 9.17　对应于图 9.16 中给出的权重系数的窗函数

9.3.3 编码参数选择

在9.3.1节中所描述的混合编码器中，编码器必须做出多种选择，包括在一个帧间编码帧中每个宏块所用的模式、每个宏块的QP、运动估计方法和参数（例如，重叠或不重叠、块尺寸、搜索范围），等等。这些编码参数的每一种组合都会产生编码视频的总码率与失真之间不同的折中。在本小节中，我们讨论如何选择这些参数的问题。

在混合编码的早期发展中，这些决策通常是启发式做出的。例如，在帧间编码图像中，基于宏块本身的方差 σ^2_{intra} 和运动补偿误差的方差 σ^2_{inter} 可以决定一个宏块是用帧内模式还是帧间模式编码[①]。如果 $\sigma^2_{\text{intra}} \leqslant \sigma^2_{\text{inter}} + c$，那么就选择帧内模式而不选择帧间模式。这种启发式决策源于下述观点：在给定的失真下一个块编码所需要的比特数与块的方差成正比。加入正的常数 c 是为了把帧间模式中运动矢量编码所需要的附加比特数计算在内。

为了改善性能，不同参数之间的选择可以由率失真最优化方法确定。这里，采用不同参数所需要的比特以及产生的失真（例如，MSE），是通过用这些参数设置对信源实际地进行编码确定的。在码率与失真之间产生最佳折中的设置被随之选定。一般，存在编码序列平均码率的约束，问题是寻找具有最小失真、同时满足码率约束的参数设置。这个有约束的最优化问题可以用拉格朗日乘子法[7]或动态编程法[2]解决。

让我们考虑一个对一帧中的所有宏块确定编码模式的例子，假设所有其他的选项是固定的，并且整个帧所期望的比特数是 R_d。用 $D_n(m_n)$ 表示第 n 个宏块采用模式 m_n 时的失真，用 $R_n(m_k, \forall k)$ 表示所需要的比特数。R_n 依赖于其他宏块的编码模式的原因在于运动矢量和DC系数是由邻块进行预测编码的。这个问题是：

$$\begin{aligned} &\text{最小化} \quad \sum_n D_n(m_n) \\ &\text{条件是} \quad \sum_n R_n(m_k, \forall_k) \leqslant R_d \end{aligned} \tag{9.3.8}$$

用拉格朗日乘子法，这个问题转化为：

$$\text{最小化} \quad J(m_n, \forall_n) = \sum_n D_n(m_n) + \lambda \sum_n R_n(m_k, \forall_k) \tag{9.3.9}$$

其中，λ 必须满足码率约束。

严格地讲，不同宏块的最佳编码模式是相互依赖的。为了易于理解基本概念，让我们忽略码率 R_n 对其他宏块编码模式的依赖性；也就是说，我们假设 $R_n(m_k, \forall k) = R_n(m_n)$。那么每个宏块的编码模式可以通过使下式最小化单独地确定：

$$J_n(m_n) = D_n(m_n) + \lambda R_n(m_n) \tag{9.3.10}$$

如果仅有少数几个模式可供选择，那么可以通过穷尽搜索为每个块寻找最佳模式。

注意，如果 m_n（从而 R_n）是连续变量，那么最小化 J_n 就等价于设置 $\partial J_n / \partial R_n = 0$，这将导致 $\partial D_n / \partial R_n = -\lambda$。这意味每个宏块的最佳模式是在不同的宏块中产生相同的RD斜率 $\partial D_n / \partial R_n$

① 实际上，为了减少计算量，经常用每个像素值（原始的或预测误差）与块的均值的绝对差之和来代替方差。

的那个模式①。实际上,仅有有限的模式可供选择。每种可能的模式都在分段线性的 RD 曲线上对应一个工作点,从而每种模式都与一个 RD 斜率范围有关。对于给定的 λ,通过最小化 $J_n(m_n)$ 找到的不同宏块的最佳模式在它们的 RD 斜率中将具有相似的范围。

不同的宏块应该工作在相同的 RD 斜率上,这个结果是涉及参数的多重独立编码的各种 RD 最优化问题的一种特殊情况。回想在变换编码(9.1.5 节)的比特分配问题中,最佳解是使不同系数的 RD 斜率相同的解(公式(9.1.40))。在那种情况下,RD 斜率与失真成正比;因此,最佳比特分配在不同系数中产生相等的失真。

在这种方法中的一个难题是如何对给定的期望码率确定 λ。对于一个任意选择的 λ,这种方法将得到一个在特殊的码率下最佳的解,这个码率可能接近也可能不接近期望的码率。在参考文献[36]中导出了 λ 和 QP 之间的一个封闭形式的关系,它假设 QP 是基于目标比特率选择的(见下一小节)。

编码模式的率失真最佳选择是首先由 Wiegand 等考虑的[38]。在那里,为了应对同一帧和相邻帧中不同宏块的编码模式之间的相互依赖性,采用动态编程方案同时为一组宏块寻找最佳编码模式。Sullivan 和 Wiegand 的论文[36]很好地讨论了如何用类似的方法确定 H.263 编码器中的各种编码选项。值得注意的是:基于 RD 的方法优于启发式的方法,导致在 H.263 框架内相当临界的增益(大约节省 10% 的比特率或 0.5 dB 的 PSNR)[36]。实际上,考虑到相当显著的复杂度的增加,这样的增益可能并不被认为是合理的。因而,RD 最佳化方法主要是作为评价启发式方法性能的一种基准,启发式方法仍是实际的方法。

在基于 RD 的参数选择方法中,从计算上最需要的一步是收集与不同参数设定有关的 RD 数据(编码模式和 QP,以及可能的不同运动估计方法)。这将要求用所有不同的参数实际地编码所有的宏块。为了减少计算量,已经提出了一些一方面联系码率与失真,另一方面联系 QP 与编码模式的 RD 模型[5,24]。一旦获得了 RD 数据,我们可以使用拉格朗日乘子法或者使用动态编程法来求得最佳的分配。拉格朗日乘子法比较简单,但具有次最佳的性能,因为它忽略了同一帧中或相邻帧间的邻接宏块的码率之间的相关性。

除了编码参数选择外,基于 RD 的方法可以应用于图像和视频编码中的各种问题。一个重要的领域是视频编码的运动估计。传统的运动估计方法只注意使运动补偿预测误差最小化(见第 6 章),而 RD 最佳化方法还考虑对产生的 MV 进行编码所需的码率。例如,考虑到编码非零的 MV 需要额外的比特,如果把非零 MV 转换为零 MV 仅导致稍微高一点的预测误差,我们宁愿选择零 MV。而且,因为 MV 是以预测方式编码的,所以更喜欢较平滑的运动域。有关 RD 最佳运动估计的工作可以在文献[35,11,20,33,3]中找到。为更完全地了解如何用 RD 理论使图像和视频编码优化,读者可以参考 Ortega 和 Ramchandran 的综述文章[58],其中也包含作为优化工具的拉格朗日和动态编程方法的一个很好的说明。这个课题更广泛的涵盖可以在 Schuster 和 Katsaggelos[32]中找到。

9.3.4 码率控制

视频编码中一个重要的问题是码率控制,这是指如何编码视频以使产生的比特流满足目

① 由定义,$\partial D_n/\partial R_n = \lim_{\Delta R\to 0}\Delta D/\Delta R$;也就是说,RD 斜率可以度量由每个附加比特所引起的失真下降。等斜率条件等价于要求所有的块都以这样的方式工作:在每个块中一个附加比特将使误差减少相同量。否则,一个工作在较高(较低)斜率的块应该使用较多(较少)的比特。

标比特率。在混合编码框架内,它简化为如何选择编码参数(帧率、帧间还是帧内、QP 等)以满足码率约束。在上一小节间接提到了这个问题,在那里我们描述了一个在码率约束条件下为每个宏块选择编码模式的 RD 最优化方法。

为获得满意的视觉,所表示的视频最好具有恒定(或平滑变化)的质量。因为场景的活动性(用运动以及纹理复杂度表示)随时间变化,所以需要的比特从一帧到另一帧是变化的。即使我们接受视频质量的变化,也不能严格地使比特率恒定,因为在帧的水平上使用了 VLC 和可能不同的编码模式。因此,我们可以期望的最好情况是在短的时间间隔内实现恒定的平均码率。在每个间隔内比特率的变化必须用编码器之后的一个平滑缓冲器进行处理。从编码器输出的比特首先以可变的码率进入平滑缓冲器,然后以恒定的码率移出缓冲器。最大的时间间隔和在这个间隔内所允许的变化取决于应用的延迟要求和购买得起的缓冲器的大小[①]。缓冲器越大,视频质量就越好,但也将引入更长的延迟。

码率控制通常是以三步完成的:

1. 为每个短的时间间隔(也称为码率更新间隔)更新目标平均比特率(以 bps 为单位)[②];
2. 确定编码模式(例如,I,P 或 B 帧)和在这个间隔内欲编码的每一帧的目标比特预算,它通常是基于这个间隔的目标平均码率和当前缓冲器充满度;
3. 为帧中的每个宏块确定编码模式和 QP 以满足这个帧的目标码率。

第一步取决于应用系统。对于恒定比特率(CBR)网络(例如一个调制解调线路或 ISDN 信道)上的视频传输,目标平均码率应该是恒定的。对于能容纳可变比特率(VBR)的网络,例如 ATM 网络,编码器应该试图调节码率以保持恒定的视频质量,且仍满足平均和峰值码率要求[14]。最后,对于信道容量和误码特性随时间变化的网络,例如尽力使资源的使用达到最好的因特网和无线信道,必须根据信道条件更新目标平均码率。更新间隔取决于网络条件变化有多快。在所有这些应用系统中,每个更新间隔的目标平均比特率应该不仅取决于可用带宽(在时变信道中它是需要估计的),也取决于端到端的延迟要求和当前编码器的缓冲器大小(即余下多少比特要发送)。实质上,目标码率应该设置成使得来自当前更新间隔的比特以及缓冲器中剩余的比特可以在最大的允许延迟时间内到达目的地。从这种意义上来说,码率约束是由延迟约束决定的[27]。在有损网络中,我们也必须考虑这样的事实:当某些以前已发送过的数据在传输期间受损的情况下,可能需要重传[13]。在参考文献[27]中,Ortega 提出了一种码率控制问题的分类方法,广泛讨论了如何基于网络特性和延迟要求确定目标码率。对于为因特网应用系统上的视频流设计的码率估计方法,见第 15 章中的讨论。

第二步一般是通过基于更新间隔的总码率预算选择适当的帧率,并且为间隔内所有的帧分配相同的目标码率实现的。当缓冲器占用率有溢出危险时,可以跳过一些帧。当用于前面几帧的比特大大超过规定的目标码率,或这个间隔的剩余比特预算不足以用规定的帧率对视频编码时,溢出是可能发生的。这是通常用于交互式应用系统的方法,除第一帧外,所有的帧都用帧间模式编码以维持帧间相当恒定的延迟。对于 MPEG-2 编码的视频广播或视频流,更新间隔通常

① 考虑存储器成本不断地下降,有理由假设缓冲器可以具有端到端的延迟所允许的大小,因此缓冲器大小的物理限制可以忽略[27]。编码器的缓冲器只是引起视频传输端到端延迟的一个因素。对于引起端到端延迟不同因素的描述,见第 14 章。

② 实际上,这可以用滑动窗的方式做到。也就是说,在每个新的帧期间,确定此帧之后一个短时间内的目标比特率。

对应一个图像组,包含一个 I 帧,后跟交织的 P 和 B 帧。通常,一个图像组内的码率分配,在用于 I 帧、P 帧和 B 帧的比特之间,呈现某种恒定比率[16]。更复杂的是,基于 RD 的方法试图基于场景内容在帧间分配总的比特,以便使平均失真最小。然而,所涉及到的超前准备将导致相当大的计算量和额外的编码延迟。

帧内的码率控制(即第二步)可以通过调整每个宏块的编码模式和 QP 来实现,调整可以基于启发式规则或采用 RD 最优化方法,如 9.3.3 节中所述。

上述讨论主要集中于视频传输的应用。对于存储应用系统(例如在 DVD 电影中),用于视频的总比特受到存储媒体容量限制;然而,比特率可以随时间改变,以适应场景中活动的变化。

到目前为止,所讨论的码率控制解决方案仅考虑了基于波形的编码器。对于基于物体的编码器(例如 MPEG-4),码率控制必须考虑在多个物体之间的码率分配;并且,在每个物体内,在形状、运动和纹理之间分配码率。不同物体的帧率也可以根据每个物体的重要性而改变。Vetro, Sun 和 Wang 讨论了这些问题[37]。

除了生成具有固定目标码率的比特流以外,另一种方法是产生可分级的比特流,收信机可以根据可用带宽只从这个比特流中有选择地抽取部分比特。这种方法比较适合于多个具有不同连接性能的用户访问同一个视频的场合。可分级编码在第 11 章中讨论;在第 15 章中讨论可分级视频流向具有不同带宽容量的用户发布的问题。

与码率控制有关的一个问题是码率成形。它是指在压缩层和网络传输层之间或在两个网络段之间的接口(称为码率转换器或滤波器)功能;预先压缩的视频流通过接口进行转换,使得产生的视频流的码率与可用网络带宽匹配。这个课题将在第 15 章中讨论。

重要的是要注意,在视频编码标准中,没有定义编码模式选择和码率控制。只要编码比特流符合标准的语法,编码器具有使它们的实现最优化的灵活性。

9.3.5 环路滤波*

运动补偿预测误差的信源可以分为三类:(1)运动估计误差;(2)参考帧中的噪声,它可能来源于前面几帧运动估计引起的误差以及量化误差的累积;(3)在编码帧中不是由运动引起的变化,例如遮挡和照度变化。为了从根本上抑制第二种噪声信源,可以应用环路滤波,它由应用于预测图像的低通滤波器组成。注意,对于非整数运动估计,当进行运动补偿预测时,隐含地使用了内插滤波器。类似地,环路滤波可以隐含地用 OBMC 实现。然而,如下面可以看到的,可以特别设计滤波器来降低视频信号中噪声的影响。具有滤波的运动补偿一般可以描述为:

$$\psi_p(x,y) = g(x,y) * \psi_r(x + d_x(x,y), y + d_y(x,y)) \tag{9.3.11}$$

其中,$\psi_p(x,y)$ 表示预测帧,$\psi_r(x,y)$ 表示参考帧,$g(x,y)$ 表示环路滤波器的脉冲响应,$*$ 表示二维线性卷积。下面,我们按照 Girod[10,8,9]的方法推导最佳滤波器。

为了简化设计,做以下假设:(1)参考视频帧是平稳随机场;(2)在不同空间位置上估计的运动矢量的误差是相互独立的;(3)噪声项独立于参考帧。在这些假设下,预测误差的功率谱密度(PSD)可以表示[8]为:

---

* 标有星号的节可以跳过或留待进一步研究。

$$S_{e,e}(f_x,f_y) = S_{\psi,\psi}(f_x,f_y)(1 + |G(f_x,f_y)|^2 - 2\mathcal{R}\{G(f_x,f_y)P(f_x,f_y)\}) + S_{n,n}(f_x,f_y)|G(f_x,f_y)|^2 \tag{9.3.12}$$

在这个关系式中，$S_{\psi,\psi}(f_x,f_y)$和$S_{n,n}(f_x,f_y)$分别表示参考帧的 PSD 和噪声信号；$G(f_x,f_y)$是$g(x,y)$的傅里叶变换；$P(f_x,f_y)$是运动估计误差的概率密度函数的傅里叶变换；$\mathcal{R}$表示复变量的实部。在这个分析中，假设信号是离散的，傅里叶变换是 DSFT。

对公式(9.3.12)关于$G(f_x,f_y)$求导，我们得到使均方预测误差为最小的最佳滤波器：

$$G(f_x,f_y) = P^*(f_x,f_y)\frac{S_{\psi,\psi}(f_x,f_y)}{S_{\psi,\psi}(f_x,f_y) + S_{n,n}(f_x,f_y)} \tag{9.3.13}$$

其中，* 表示复共轭。注意，若没有乘法项$P^*(f_x,f_y)$，这个滤波器就是传统的关于噪声项的维纳(Wiener)滤波器。因子$P^*(f_x,f_y)$考虑了运动估计的不精确性。

一般地，$P(f_x,f_y)$具有低通滤波特性。因此，一般来说最佳滤波器$G(f_x,f_y)$也是低通的。仔细考察公式(9.3.12)将看出$S_{e,e}(f_x,f_y)$的趋势是在低空间频率小，在高空间频率大。例如，如果对于某个高频区域$G(f_x,f_y)=0$，则对于这个频率区域预测误差将具有与参考信号相同的能量。这个分析表明，运动补偿对信号的低频分量起作用，但不对高频分量起作用，即使采用最佳滤波器也是如此。图 9.18 示出了采用和不采用最佳滤波器时两种运动估计精度水平下的$S_{e,e}(f_x,f_y)$。在两种情况下，都假设$S_{n,n}(f_x,f_y)=0$。

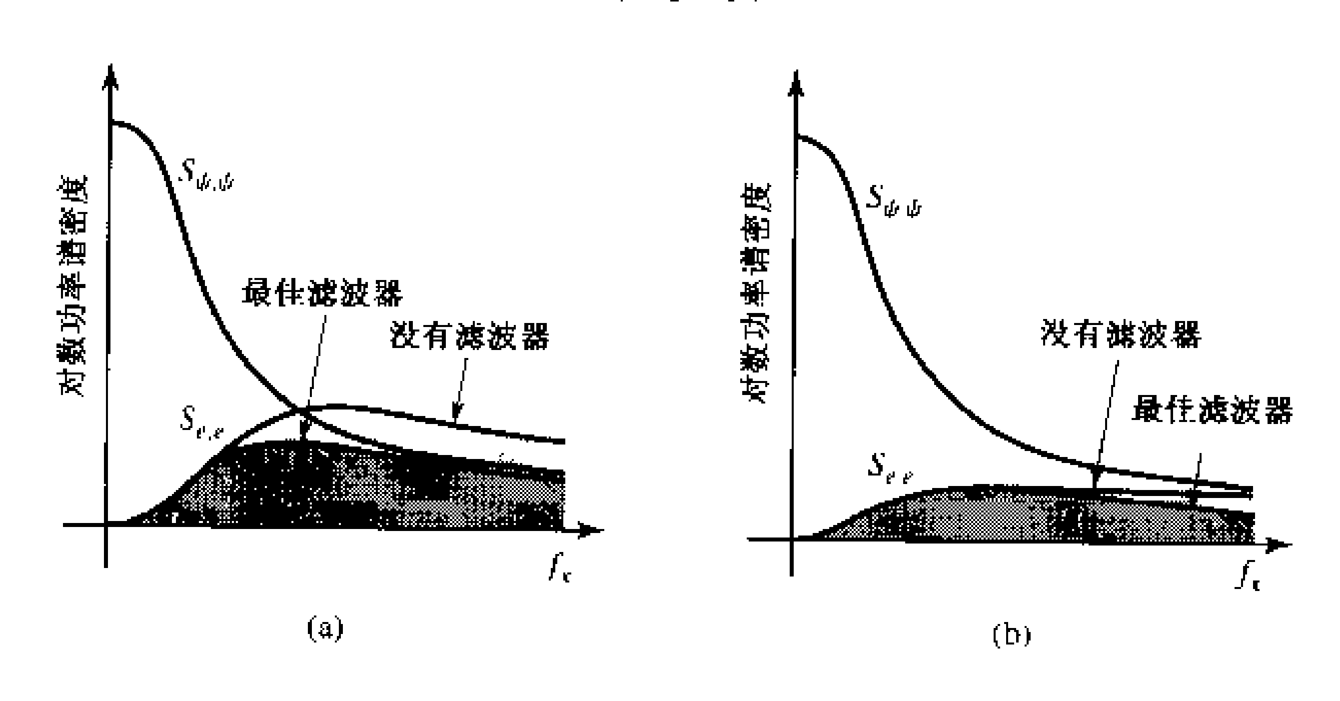


图 9.18 两种精度下的运动补偿预测误差的功率谱密度：(a)中等精度的运动估计；(b)非常精确的运动估计

已经发现环路滤波器在预测精度方面能提供显著的增益。事实上，环路滤波器的使用明确地包括进了 ITU-TH.261 视频编码标准中，这个标准采用整数像素精度的运动估计(见 13.2.1 节)。

预测误差、运动估计精度和环路滤波之间的关系 我们现在按照 Girod 的方法[8,9,10]，在采用或不采用最佳滤波的情况下，分析预测误差、运动估计精度和噪声功率之间的关系。由公式(9.3.12)中预测误差的功率谱密度，我们可以用下式确定预测误差的方差(即预测的 MSE)：

$$\sigma_e^2 = \int_{-1/2}^{1/2}\int_{-1/2}^{1/2} S_{e,e}(f_x,f_y)\,df_x df_y \tag{9.3.14}$$

假设信号功率谱是各向同性的，具有方差σ_s^2，噪声是具有方差σ_n^2的平坦的功率谱，运动补偿

误差服从高斯分布,具有方差 $\sigma_{\Delta d}^2$;Girod 为各种运动估计精度和信噪比计算了预测误差的方差。图 9.19 示出了他的分析结果。最有趣的看法是:存在一个运动估计精度的临界点,超过这个点进一步改善预测精度的可能性是很小的。这个临界点随信号中噪声电平的增加而增加。也就是说,当信号存在很强的噪声时,不存在一个能够很准确地估计运动的点。即使用非常准确的运动估计,由于噪声的存在,预测帧仍然易于产生误差。这幅图也揭示出,当噪声电平很高时,维纳滤波可能非常有用,但在低噪声的情况下并不十分有效。

在图 9.19 中,运动估计误差的方差与像素的搜索精度有关。这个关系的建立是由于假设运动估计误差完全是由于对给定的小数精度进行四舍五入引起的。垂直虚线指示用图上所标示的运动补偿精度可以达到的最小位移误差的方差。我们可以看到,在低噪声情况下,需要不大于 1/8 像素的精度,而在高噪声情况下,1/2 像素精度就足够了。这些分析结果已被实际试验结果所确认[9]。已经提出的一种观点是,对于高质量广播视频信号,1/4 像素精度的运动估计对一个实际的编码器是足够的,而对视频电话型信号,1/2 像素精度似乎是一个期望的界限。在这两种情况下,都比整数像素精度有显著的增益。

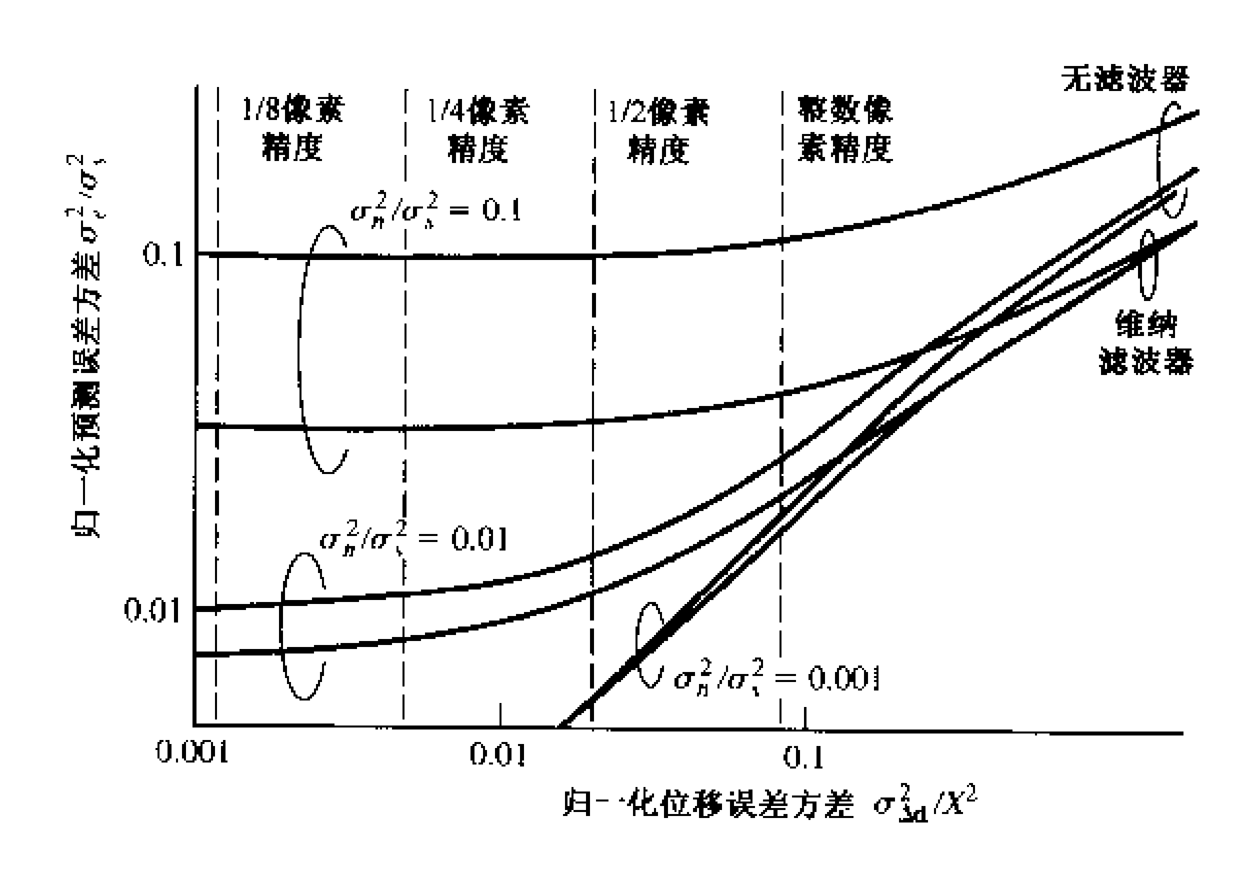


图 9.19　运动补偿精度对有噪信号的预测误差方差的影响。X 表示一个像素的宽度(假设高等于宽)

9.4　小结

变换编码(9.1 节)

- 变换提供一种把一组样点(即一个图像块)表示为基本模式的线性组合的方法。变换编码的目的是去除原始样点的相关性,并把能量压缩到少数几个系数上,以便能有效地应用标量量化。在这些准则下的最佳变换是 KLT。
- 为了从变换编码中得到最大的增益,必须适当地在系数之间分配比特。最佳分配(公式(9.1.43))使不同系数遭受的失真均衡化(公式(9.1.44))。最终的失真取决于系数方差

的几何平均。

- KLT 取决于信号的二阶统计特性，因此，它是依赖于图像的，且难于计算。对于大多数图像信号，DCT 是 KLT 很好的近似，并且可以用快速算法计算它。
- 基于 DCT 的编码方法(9.1.7 节)是图像编码的有效方法，并且已在图像和视频编码的国际标准中采用。
- 量化的最佳化和二进制编码过程在变换编码器中是非常重要的。对于相似的量化和二进制编码方案，DCT 和小波变换导致相似的编码效率。

预测编码(9.2 节)

- 预测的目的也是要减少欲编码样点之间的相关性，以便可以有效地应用标量量化。预测器应该设计成使预测误差最小。通过解尤里 - 沃克方程(公式(9.2.6))可以确定 LMMSE预测器。
- 为了避免编码器中用于预测的参考样点与解码器中所用的参考样点之间的失配，需要闭环预测；在闭环预测中编码器必须重复与解码器相同的操作。
- 理论上，当预测器阶数与变换长度相同时，预测编码优于变换编码。然而，预测编码产生的比特流对传输误码很敏感。
- 对于视频编码，预测可以在空间域和时间域进行。在时间方向上，考虑物体运动的影响需要进行运动补偿。仅用很低阶的预测器(通常仅用前面的一帧)以维持合理的复杂度，并减小传输误码扩散的影响。

基于块的混合视频编码(9.3 节)

- 基于块的混合编码器(9.3.1 节)有效地联合了运动补偿预测和变换编码。因为它具有相对较低的复杂度和好的编码效率，所以它在各种视频编码的国际标准中都得到采用(第 13 章)。
- 在混合编码的框架内，适当地进行运动估计和补偿(例如，有或没有重叠，固定的或可变的块尺寸)和选择操作模式(帧内或帧间模式，预测模式、等等)可以改善性能。RD 最优化方法选择这些编码选项，以使在给定码率约束下使失真最小(9.3.3 节)。
- 码率控制(9.3.4 节)在视频编码中是一个重要问题。特别对于实时应用系统，过大的码率会导致由传输延迟和丢码引起的质量退化(改善的反面)。混合编码器中的码率控制通常是通过调整帧率和量化参数实现的，采用启发式或者 RD 最优化方法。
- 编码模式选择和码率控制可以显著地影响编码器的性能。注意，所有采用混合编码框架的视频编码标准仅定义了比特流语法，从而仅定义了解码器操作。这给了编码器在最优化其操作上的灵活性，包括运动补偿、编码模式选择和码率控制。这些一般是区分所有符合相同标准的视频编码系统的因素。除编码效率的考虑以外，其他可以区分系统的因素是，编码器如何使比特流对传输误码更有复原能力(通过在标准框架内明智地选择编码参数)，以及解码器如何反抗传输误码。这些问题将在第 14 章中讨论。
- 通过应用环路滤波(9.3.5 节)可以改善运动补偿精度；它抑制前面帧中量化误差的扩散。可以达到的运动补偿精度不仅取决于运动估计精度，也取决于所用的环路滤波器和信号中的噪声电平(9.3.5 节)。

9.5　习题

9.1　证明公式(9.1.11 ~ 9.1.16)中的关系。

9.2　证实一维 DCT 的基函数形成正交基。

9.3　证明两个正交的一维基的外积所形成的二维基也是正交的。

9.4　对于特殊情况 $\rho_h = \rho_v = \rho_d = \rho$ 重做例 9.4。

9.5　对于特殊情况 $\rho_h = \rho_v = \rho_d = \rho$ 重做例 9.5。把 KLT 变换基与 DCT 进行比较。

9.6　对于特殊情况 $\rho_h = \rho_v = \rho_d = \rho$ 重做例 9.6。

9.7　考虑一个预测图像编码系统，每个像素是由一个在上方、一个在左侧的两个相邻像素预测的。假设图像统计特性与例 9.4 中所考虑的一样。导出最佳预测器和预测编码增益。把这个结果与例 9.6 中的相比较。哪一个预测器的效率更高？

9.8　写出考察用部分 DCT 系数近似图像的效果的 C 或 MATLAB 代码。采用 8×8 DCT，用 $K < 64(K = 4,8,16,32)$个系数重建图像。为提供一个满意的重建需要多少系数？

9.9　写出考察 DCT 域中量化效应的 C 或 MATLAB 代码。采用 8×8 DCT，选择量化步长是图 9.7 中所给的量化表的倍数。尝试用 0.5,1,2,4,8 和 16 的缩放因子。仍能提供一个满意图像的最大缩放因子是多少？

9.10　两种简单的变换编码方法是：(a)仅保留前 K 个系数(称为区域编码)；(b)仅保留超过一个特定阈值的系数(称为阈值编码)；讨论这两种方法的优劣。

9.11　考虑以下三种视频编码方法：直接编码一帧；编码每两帧之间的差值；在每两帧间进行基于块的运动估计并编码运动补偿误差图像。回想达到一个给定的失真所需要的比特率与欲编码的信号的方差成正比。从一个序列中取两个相邻的视频帧，计算并比较以下三种情况的方差(a)直接的一帧；(b)两帧间直接的差值；(c)运动补偿误差图像。基于你的结果，你能确定这三种编码方法中哪一种效率更高吗？

9.12　方差只是所需要的比特率的粗略指示。如果我们对习题 9.11 中三种方法的每一种都应用 DCT 编码，比特率的一个较精确的测度是用所选择的量化参数量化后非零的 DCT 系数的数目。写出进行这个实验的 C 或 MATLAB 代码。

9.13　证明 OBMC 最佳线性估计器的解是如公式(9.3.4)所给出的。

9.14　证明对应于图 9.16 所给出的加权系数的窗函数是如图 9.17 所给出的。以图 9.15 的形式画出它来，并评论两个窗函数之间的相似性和差别。

9.15　写出实现一个基于块的混合编码器的基本形式的 C 或 MATLAB 程序。为简单起见，仅考虑帧内模式和 P 模式。除了整个用帧内模式编码的第一帧外，根据是原始的宏块还是运动补偿误差有较小的方差，后面帧中的每个宏块将用帧内模式或者用 P 模式编码。对于运动估计，可以选择用 EBMA 或者用一些快速算法实现。对于无损编码部分，可以使用标准中给出的哈夫曼表(例如，参考文献[17])，或者通过计算欲编码的符号的熵来估计所需要的比特率。把该程序应用于几个具有不同运动水平、采用不同参数(QP、帧率、内帧或帧内块率，等等)的测试序列。评价码率、失真和复杂度(执行时间)之间的折中。播放解压缩视频，观察具有不同 QP 的编码效应。

9.6 文献目录

[1] Ahmed, N., T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Trans. Computer* (1974), 23:90–93.

[2] Bellman, R. *Dynamic Programming*. Princeton NJ: Princeton University Press, 1957.

[3] Chen, M. C., and A. N. Willson. Rate-distortion optimal motion estimation algorithms for motion-compensated transform coding. *IEEE Trans. Circuits Syst. for Video Technology* (April 1998), 8(2):147–58.

[4] Chen, W.-H., et al. A fast computational algorithm for the discrete cosine transform. *IEEE Trans. Commun.* (Sept. 1977), COM-25:1004–09.

[5] Chiang, T., and Y.-Q. Zhang. A new rate control scheme using quadratic rate-distortion model. *IEEE Trans. Circuits Syst. for Video Technology* (Feb. 1997), 7:246–50.

[6] Clark, R. J. *Transform Coding of Images*. London: Academic Press, 1985.

[7] Everett, H. Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. *Operational Research* (1963), 11:399–417.

[8] Girod, B. The efficiency of motion-compensating prediction for hybrid coding of video sequences. *IEEE J. Select. Areas Commun.* (1987), SAC-5:1140–54.

[9] Girod, B. Motion-compensating prediction with fractional-pel accuracy. *IEEE Trans. Commun.* (1993), 41:604–12.

[10] Girod, B. Motion compensation: Visual aspects, accuracy, and fundamental limits. In M. I. Sezan and R. L. Lagendijk, eds., *Motion Analysis and Image Sequence Processing*, Boston: Kluwer Academic Publishers, 1993, 126–52.

[11] Girod, B. Rate-constrained motion estimation. In *SPIE Conf. Visual Commun. Image Processing* (Sept. 1994), 1026–34.

[12] Gonzalez, R. G., and R. E. Woods. *Digital Image Processing*. 2d ed. Reading, MA: Addison-Wesley, 1992.

[13] Hsu, C.-Y., A. Ortega, and M. Khansari. Rate control for robust video transmission over burst-error wireless channels. *IEEE J. Select. Areas Commun.* (May 1999), 17(5):756–73.

[14] Hsu, C.-Y., A. Ortega, and A. R. Reibman. Joint selection of source and channel rate for VBR video transmission under ATM policing constraints. *IEEE J. Select. Areas Commun.* (Aug. 1997), 15:1016–28.

[15] ISO/IEC. IS 10918-1: Information technology—digital compression and coding of continuous-tone still images: Requirements and guidelines, 1990. (JPEG).

[16] ISO/IEC. MPEG-2 Test Model 5 (TM5). Public document, ISO/IEC JTC1/SC29/WG11/93-225b, April 1993.

[17] ITU-T. Recommendation H.263: Video coding for low bit rate communication, 1998.

[18] Jain, A. K. *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[19] Jayant, N. S., and P. Noll. *Digital Coding of Waveforms*. Englewood Cliffs, NJ:

Prentice-Hall, 1984.

[20] Kossentini, F., Y.-W. Lee, M. J. T. Smith, and R. Ward. Predictive RD optimized motion estimation for very low bit rate video coding. *IEEE J. Select. Areas Commun.* (Dec. 1997), 15(9):1752–63.

[21] Lee, B. G., A new algorithm to compute discrete cosine transform. *IEEE Trans. Acoust., Speech, Signal Processing* (Dec. 1984), ASSP-32:1243–45.

[22] Li, W. Vector transform and image coding. *IEEE Trans. Circuits Syst. for Video Technology* (Dec. 1991), 1(4):297–307.

[23] Li, W. and Y.-Q. Zhang. Vector-based signal processing and quantization for image and video compression. *Proceedings of the IEEE* (Feb. 1995), 83(2):317–33.

[24] Lin, J.-L., and A. Ortega. Bit rate control using piecewise approximated rate-distortion characteristics. *IEEE Trans. Circuits Syst. for Video Technology* (Aug. 1998), 8:446–59.

[25] Nogaki, S., and M. Ohta. An overlapped block motion compensation for high quality motion picture coding. *IEEE Int. Conf. Circuits and Systems* (May 1992), 184–87.

[26] Orchard, M. T., and G. J. Sullivan. Overlapped block motion compensation: An estimation-theoretic approach. *IEEE Trans. Image Process.* (1994), 3:693–99.

[27] Ortega, A. Variable bit rate video coding. In M. T. Sun and A. R. Reibman, eds., *Compressed Video over Networks,* New York: Marcel Dekker, 2001.

[28] Ortega, A., and K. Ramchandran. Rate-distortion methods for image and video compression. *IEEE Signal Processing Magazine* (Nov. 1998), 15:23–50.

[29] Pennebaker, W. B., and J. L. Mitchell. *JPEG Still Image Data Compression Standard,* New York: Van Nostrand Reinhold, 1993.

[30] Rabiner, L. R., and R. W. Shafer. *Digital Processing of Speech Signals.* Englewood Cliffs, NJ: Prentice Hall, 1978.

[31] Ramstad, T. A. Considerations on quantization and dynamic bit allocation in subband coders. *IEEE Int. Conf. Acoust., Speech, Signal Processing* (April 1986), 841–44.

[32] Schuster, G. M., and A. K. Katsaggelos. *Rate-distortion Based Video Compression.* Boston: Kluwer Academic Publishers, 1997.

[33] Schuster, G. M., and A. K. Katsaggelos. A video compression scheme with optimal bit allocation among segmentation, motion, and residual error. *IEEE Trans. Image Process.* (Nov. 1997), 6:1487–501.

[34] Segall, A. Bit allocation and encoding for vector sources. *IEEE Trans. Inform. Theory* (March 1976), IT-22:162–69.

[35] Sullivan, G. J., and R. L. Baker. Rate-distortion optimized motion compensation for video compression using fixed vs. variable size blocks. In *Global Telecomm. Conf.* (GLOBECOM '91) Dec. 1991, 85–90.

[36] Sullivan, G. J., and T. Wiegand. Rate-distortion optimization for video compression. *IEEE Signal Processing Magazine* (Nov. 1998), 15:74–90.

[37] Vetro, A., H. Sun, and Y. Wang. MPEG-4 rate control for multiple video objects. *IEEE Trans. Circuits Syst. for Video Technology* (Feb. 1999), 9:186–99.

[38] Wiegand, T., M. Lightstone, D. Mukherjee, T. Campell, and S. K. Mitra. Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard. *IEEE Trans. Circuits Syst. for Video Technology* (April 1996), 6(2):182–90.

第 10 章　基于内容的视频编码

在 8.1 节中,我们给出了不同视频编码系统的概述。区分编码系统的一种方法是比较它们的信源模型。在第 9 章中描述的视频编码是基于不自适应视频序列中运动物体形状的信源模型的。在本章中,我们主要介绍考虑视频序列中物体的形状的视频编码算法。为了有效地传输任意形状的视频物体,必须编码物体的形状以及纹理。在 10.1 节中,我们描述编码一个物体的二维形状的几种方法。10.2 节给出有效编码区域纹理的算法。联合编码形状和纹理的两种算法在 10.3 节中简单介绍。在 10.4 节和 10.5 节中,我们描述这些技术如何分别应用于基于区域和基于物体的编码。基于物体的编码可以基于二维和三维物体模型。用三维物体模型,可以把基于物体的编码推广到基于知识的和语义的编码,如 10.6 节和 10.7 节所述。在 10.8 节中,我们提出一种概念,允许把传统的基于块的混合编码与基于物体的、基于知识的和语义编码一体化。

10.1　二维形状编码

假定第 k 个物体的二维形状是通过阿尔法映射 M_k 定义的:

$$M_k = \{m_k(x,y) | 0 \leqslant x \leqslant X, 0 \leqslant y \leqslant Y\},\quad 0 \leqslant m_k \leqslant 255 \tag{10.1.1}$$

形状 M_k 对于每个像素 $\mathbf{x}=(x,y)$ 定义了它或者属于该物体($m_k(\mathbf{x})>0$),或者不属于该物体($m_k(\mathbf{x})=0$)。对于一个不透明的物体,对应的阿尔法值是 255,而对于透明物体,它们的范围从 0 到 255。通常,阿尔法映射与视频序列中的亮度信号一样有相同的空间和时间分辨率。在视频编辑应用系统中,用阿尔法映射来描述物体形状和物体的透明度。让我们假设有一个背景图像 $\psi_b(\mathbf{x})$,图像 $\psi_o(\mathbf{x})$ 表示物体,$m_o(\mathbf{x})$ 表示阿尔法映射。把物体覆盖在背景上是按照下式完成的:

$$\psi(\mathbf{x}) = \left(1 - \frac{m_o(\mathbf{x})}{255}\right)\cdot\psi_b(\mathbf{x}) + \frac{m_o(\mathbf{x})}{255}\cdot\psi_o(\mathbf{x}) \tag{10.1.2}$$

阿尔法映射的幅度确定物体的可见度有多大。在本章中,我们仅涉及二进制的物体形状;也就是说,$m_k(\mathbf{x}) \in \{0,255\}$。

有两类二进制形状编码器。基于位图的编码器(10.1.1 节)对每个像素属于还是不属于物体进行编码。基于轮廓的编码器(10.1.2 节)对物体的轮廓进行编码。为了恢复出物体形状的位图,轮廓用物体标号填充。在纹理随着形状信息传输的情况下,可以用隐含的形状编码器;通过它可以从纹理导出形状信息(见 10.3 节)。如同纹理编码,二进制形状可以用有损或无损的方式编码。因此,重要的是定义有损编码形状的质量测度(10.1.3 节)。

10.1.1　位图编码

基于位图的形状编码器规定一个二进制阿尔法映射。在其最简单的形式中,我们扫描阿尔法映射并根据每个像素是否属于物体为它传输 0 或 1。这样的算法效率低,因为它没有考

虑相邻像素之间的统计相关性。在本节中,我们描述三种提高编码效率的方法:一类算法是以扫描行的顺序扫描位图并确定黑像素和白像素的游程;第二类是基于相邻像素的彩色,编码位图的每一个像素;第三类是把形状定义为基本形状的树,这些基本形状的排列完全填充物体的形状。

行扫描编码　在这种方法中,我们逐行地扫描图像,并用可变长编码器传输黑白像素的游程,由此利用像素之间的一维相关性。这种方法被用于传真标准 G3[40]。随后,我们讨论这种编码器的一个推广,它同时考虑了相邻扫描行之间的相关性。

用于传真标准 G4[23] 和 JBIG[24] 中的修正的 READ (相对元素地址指定,relative element address designate)码,参考前面的扫描行编码每个扫描行,由此利用二维的相关性。该算法扫描文件的每一行,并对变化像素的位置进行编码,在变化像素处,扫描行的彩色发生变化(图 10.1)。在这个逐行方案中,当前行每个变化像素的位置是相对于紧挨着当前行的参考行的相应变化像素,或者当前行中前一个变化像素的位置编码的[40]。图 10.1 用来解释修正 READ 编码器的一个实现。我们假设已经编码了块的前五行,因此编码器知道在当前块中像素 $a0$ 和 $b1$ 的位置。在物体边界上的未知点 $a1$ 参考这两个像素 $a0$ 和 $b1$ 进行编码。像素 $a0$ 是最后一个在 $a1$ 前编码的边界像素。像素 $b1$ 是 $a0$ 上一行、在 $a0$ 的右边并与 $a0$ 的彩色相反的第一个变化像素(如果这样的点存在的话)。如果不存在这样的点,那么 $b1$ 是与 $a0$ 同行的最左边的变化像素。为了编码 $a0$ 与 $a1$ 之间的距离,选择三种模式之一:垂直的、水平的或垂直通过的。假设所有的像素以光栅扫描的顺序被编号,块的左上角的号码为 0 (例如,图 10.1 中 $\mathrm{num}(a0)=34$),并且对列从左到右编号(例如,$\mathrm{col}(a0)=2$),则按照下式选择模式:

$$\text{模式}=\begin{cases}\text{垂直的} & \text{如果}\ |\mathrm{col}(a1)-\mathrm{col}(b1)|\leqslant T\\ \text{水平的} & \text{否则,如果}\ \mathrm{num}(a1)-\mathrm{num}(a0)<W\\ \text{垂直通过的} & \text{否则}\end{cases}\qquad(10.1.3)$$

其中 T 是阈值,W 是欲编码的块的宽度。T 值取决于扫描行的长度,可以低到 $T=5$。在图 10.1 的例子中,如果 $T=5$,则 $a1$ 以垂直模式编码。

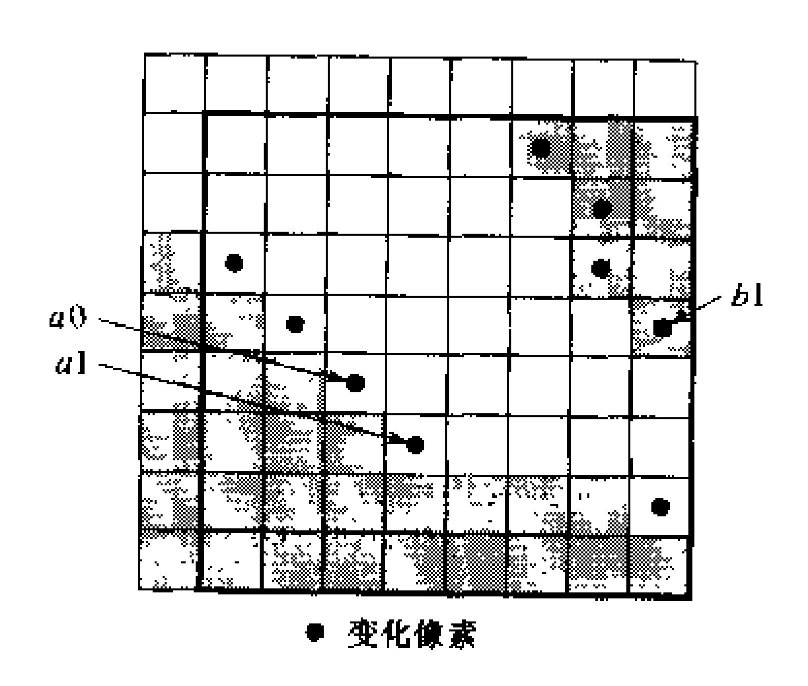


图 10.1　对于修正的 READ 编码器,变化的像素定义物体的边界

在垂直模式中,距离 $\mathrm{col}(a1)-\mathrm{col}(b1)$ 用 8 个 VLC 表之一进行编码,按照物体边界的方向选择表,方向是由一个位于像素 $b1$ 上面的模板所定义的(图 10.1)。

在水平模式中,$a1$ 的位置作为它到 $a0$ 的距离被编码。正是由于选择了水平的而不是垂

直的模式，解码器有时可以推导出 $a0$ 与 $a1$ 之间的最小距离。在这种情况下，仅编码关于这个最小距离的差值。

当 $a0$ 之后至少有一行的点没有边界像素时，按照公式(10.1.3)应该选择垂直通过模式。在这种情况下，我们为每一个没有物体边界的行发送一个码字。最后一个码字编码距物体边界下一个点的剩余距离。

像素方式编码 这种编码器利用欲编码的二进制形状信息的空间冗余。像素以扫描行顺序逐行进行编码。每个像素的彩色采用条件熵编码(见 8.3.1 节)。一个 n 像素的模板用来为预测当前像素的彩色定义上下文。图 10.2 给出了一个为编码当前像素用 10 个像素定义上下文的模板。模板在欲编码像素的左、右、上方各扩展两个像素[5]。这个上下文(即模板中像素的彩色)决定了我们用于编码当前像素的代码。因为每个像素只能具有两种彩色中的一种，所以总共有 $2^{10}=1024$ 种不同的上下文。对于一个给定的上下文，由于每个像素只能从两种可能的彩色中选择，用哈夫曼编码将不会使编码效率增加，这是因为哈夫曼编码有 1 个比特的最小码字长度。然而，我们可以用算术编码器以少于 1 个比特编码该像素的彩色(8.4.2 节)。对于每个上下文，算术编码器用不同的概率表。对于图 10.2 的模板，我们将会需要 1024 个概率表。假设必须存储一种符号(比如，黑色)的概率，且可以计算出其他的符号(比如，白色)的概率，那么我们只需存储 1024 种概率。

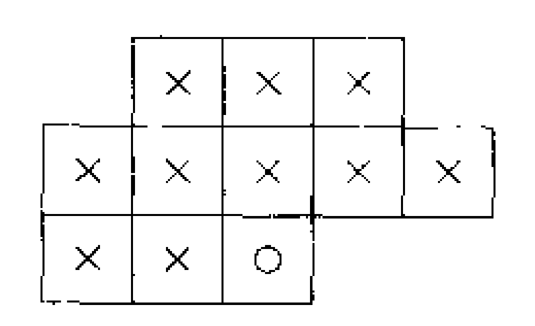


图 10.2 定义欲编码像素(由"O"表示)的上下文的模板

在 MPEG-4 的形状编码方法中，图像被分割成正方形的块。共分为三种不同类型的块：透明的和不透明的块，以及在物体边界上含有透明和不透明的像素的边界块。边界块是用上述方法编码的，这种编码方法称为基于上下文的算术编码[30]。

四叉树编码 四叉树通过把不同大小的不重叠的方块放入形状内来描述物体形状，以便尽可能准确地描述形状。通常，方块的大小是 2 的幂。我们定义块的最大尺寸和最小尺寸。如果最小方块的尺寸是一个像素，我们就可以准确地描述形状。第一步，把最大尺寸 $M\times M$ 像素的方块一个挨一个地放置在图像中。然后，基于均匀性准则决定是否需要进一步分割一个方块。如果是的话，那么就要用尺寸为 $M/2\times M/2$ 的四个方块替换这个方块。递归地重复这个过程(图 10.3)。根据均匀性准则和最小的方块尺寸，这个过程生成一个精确的或近似的形状逼近。均匀性准则定义在任何给定的方块中允许错误表示的像素的个数 n。这个数可以是固定的，也可以作为当前被考虑的方块尺寸的函数而变化。

可以用二进制符号编码四叉树。我们以深度优先的方式遍历树。在图 10.3 的例子中，1 表示相关的方块不再进一步分割，0 表示该方块被进一步分割，并且接下来的四个符号表示这个方块的子方块的状态。递归地重复这样的描述。描述完这个树以后，我们必须对每个方块指明它是否属于该物体。这是通过为树中的每个端节点发送一个二进制符号实现的。在图 10.3 中，我们用 1 表示属于物体的方块。因此，这个形状码需要与在四叉树码中所得到的 1 一

样多的符号。对形状码使用多级符号使我们仅用一个四叉树就能描述一个图像内的几个物体(见习题 10.6)。

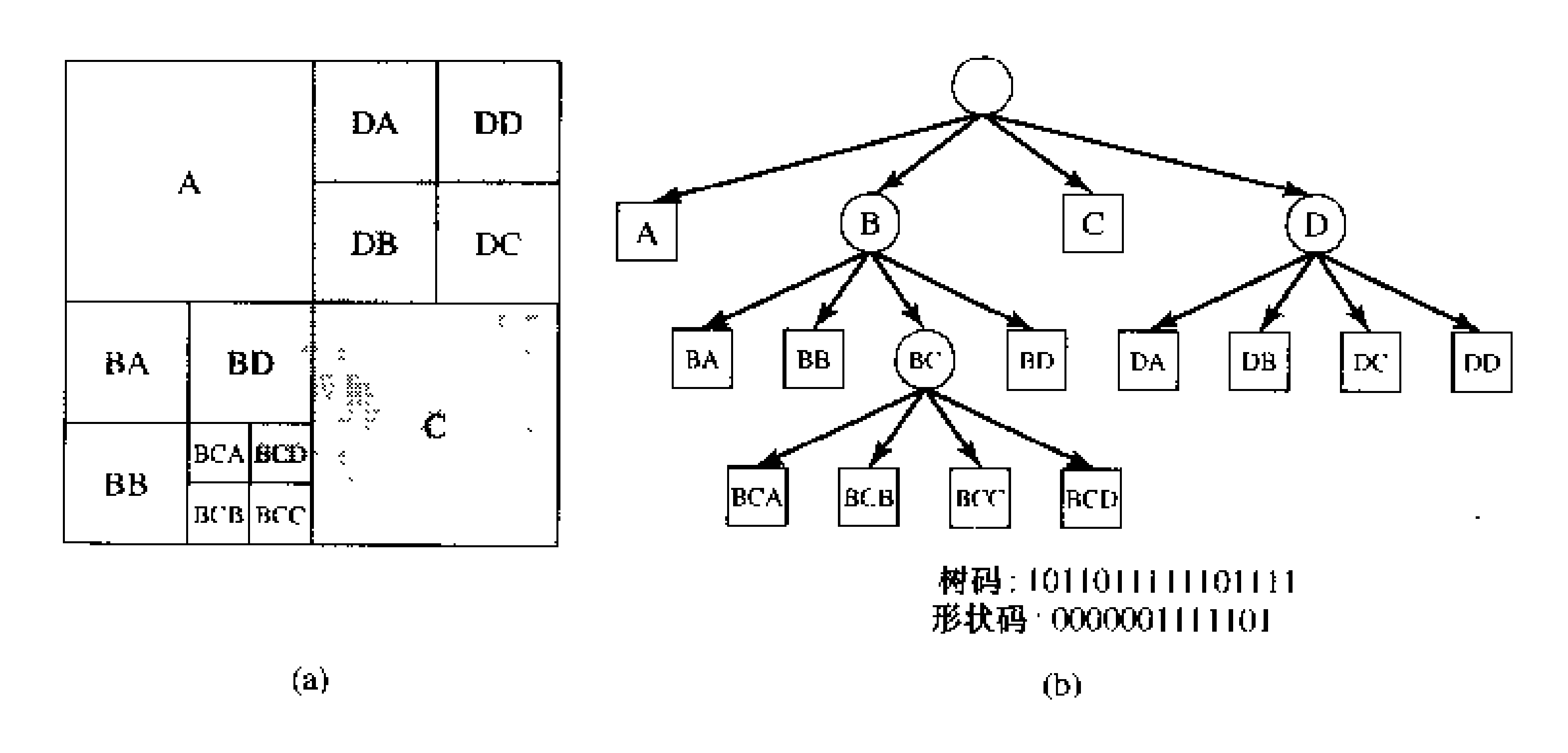


图 10.3　(a) 一个图像被分割成一个描述物体形状的四叉树；(b)带有描述该树的码和每个方块标号的相关四叉树，由此可以重建形状

10.1.2 轮廓编码

目前已经发表了许多关于基于轮廓的形状表示和编码方面的文章。不同的应用发展了这一研究：对于物体边界的无损和有损编码，开发了链编码器[14,11]以及多边形和高阶的逼近[18,54,43,15]。为了识别的目的，开发了如傅里叶描述符那样的形状表示，使平移不变、旋转不变以及缩放不变的形状表示成为可能[61]。

链编码　链码跟踪物体的轮廓。编码物体边界上的起点坐标后，链码对下一个边界像素所处位置的方向进行编码(图 10.4)。链码可以编码轮廓的方向(直接链码)或方向的变化(微分链码)[9]。微分链码的效率较高，因为它们利用了相继链接之间的统计相关性。算法以它们考虑一个像素是有 4 或 8 个相邻像素(对于矩形格栅)还是有 6 个相邻像素(对于六边形格栅)来区别。在图 10.4 中，我们定义边界像素是物体的一部分；然而，这会引起“双重形状”问题——例如，图 10.4 中的白色像素有它们自己的一组边界像素。当两个物体相接触时，这种边界像素的定义会引起冗余信息编码。有几种定义像素之间物体边界的算法可避免这种问题的发生[52]。链码是无损形状编码器；也就是说，它们准确地描述给定的轮廓。然而，我们可以在用链码编码前对轮廓进行预处理(例如平滑或量化)。有时，这种预处理并不是与链编码器分开考虑的；在那样的情况下，链编码可以认为是有损编码器。

Freeman [14] 最早提出了将链编码用于边界量化和编码，这在过去 40 多年中已引起了相当多的注意[50, 33, 41, 29]。曲线是用格栅交点方案量化的 [14]，量化后的曲线用一串增量表示。由于假设平面曲线是连续的，格点之间的增量被限制于 8 个相邻的格点，因此可以用三个比特表示增量。对于边界形状的无损编码，需要每边界像素 1.2 和 1.4 比特之间的平均码率[11]。

对这种基本方案已经有了许多扩展，如广义链码[50]，通过使用不同长度和不同角度分辨率的链接改善了编码的效率。在参考文献[29]中，提出了使用链码串中的图形来增加编码效

率的方案。对链码的理论性能也已经产生了兴趣。在参考文献[33]中,比较了不同量化方案的性能,而在参考文献[41]中,研究了具有预处理的某种链码的率失真特性。

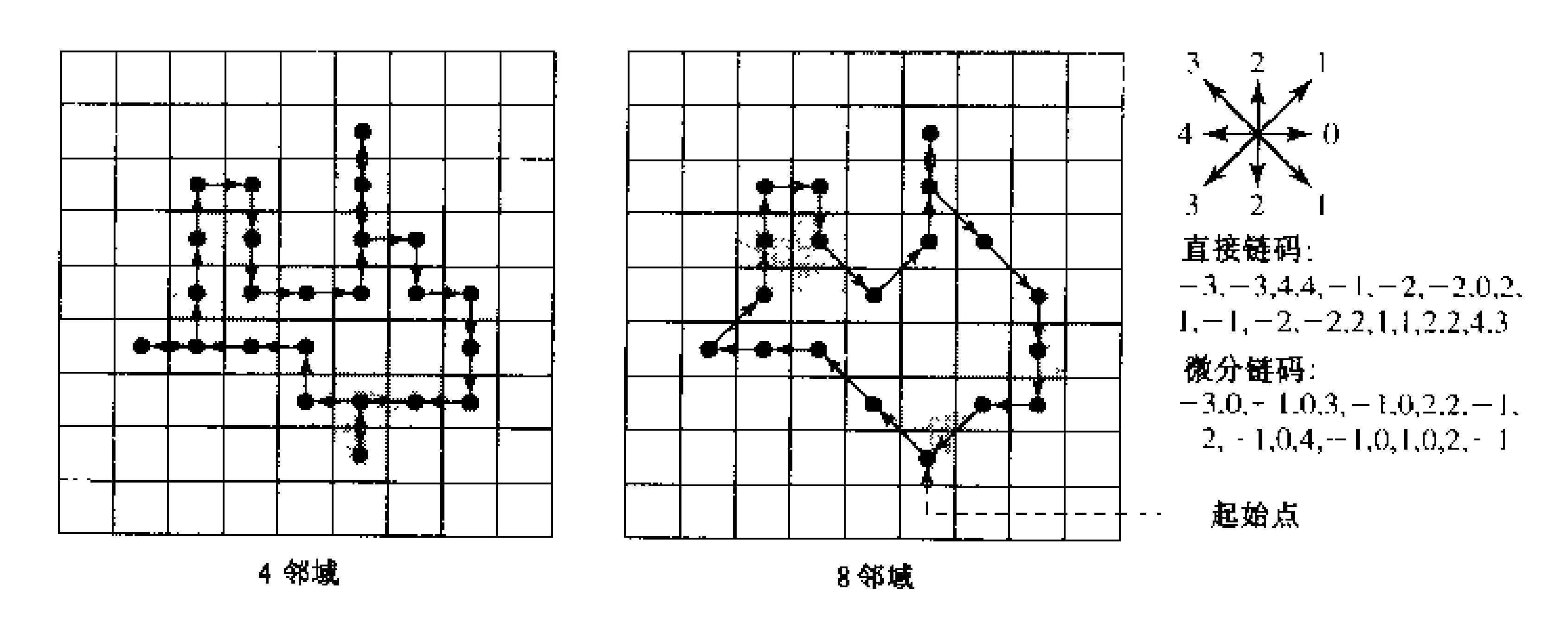
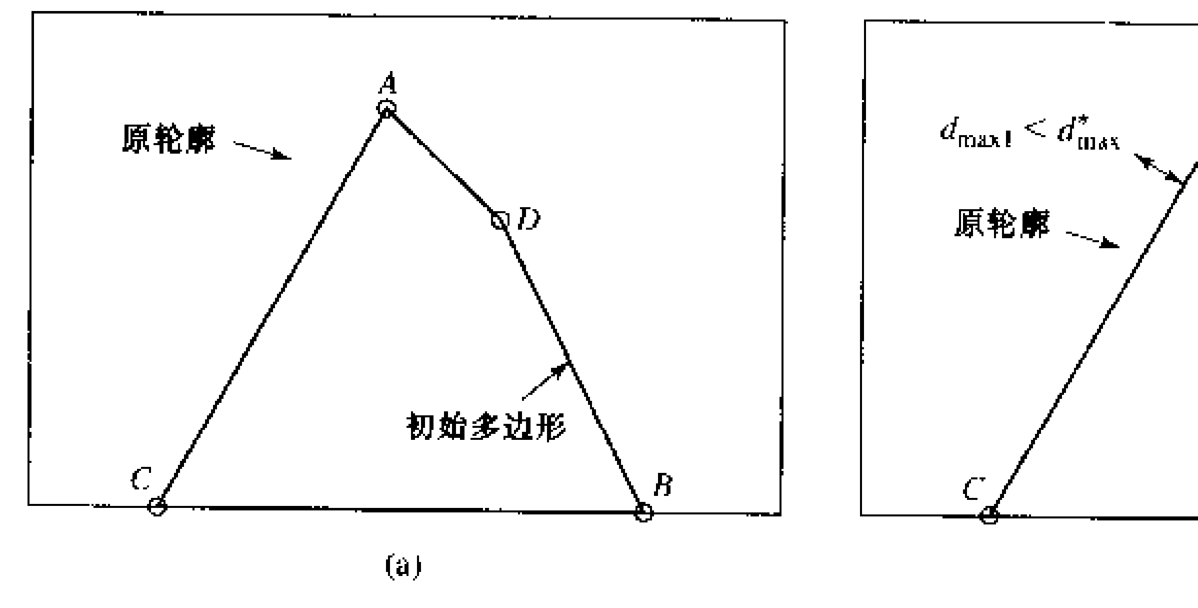


图 10.4 具有 4 邻域和 8 邻域像素的链码。我们给出了 8 连接链的直接链码和微分链码的例子。这两种码的第一个符号是相同的,定义了起始方向。微分码的后续符号是通过把最近的已编码符号的方向对准方向星的0方向生成的

为了提高编码效率,一些链码也含有轮廓的简化[42]。这类似于用形态学滤波器滤波物体的形状,然后用无损链码编码。简化相当于图像编码中的幅度量化。链码限制于无损形状表示,而以下的形状编码算法允许有损形状表示。

多边形逼近 对基于物体的分析与合成编码开发了基于多边形的形状表示[18,19]。物体形状是用连续的线段链逼近的(图 10.5)。从直觉上,我们可以设想多边形表示只用几个线段来描述具有直边缘的几何物体是很有效的。曲线的边界将需要许多的线段,以便给出一个看起来很自然的边界。



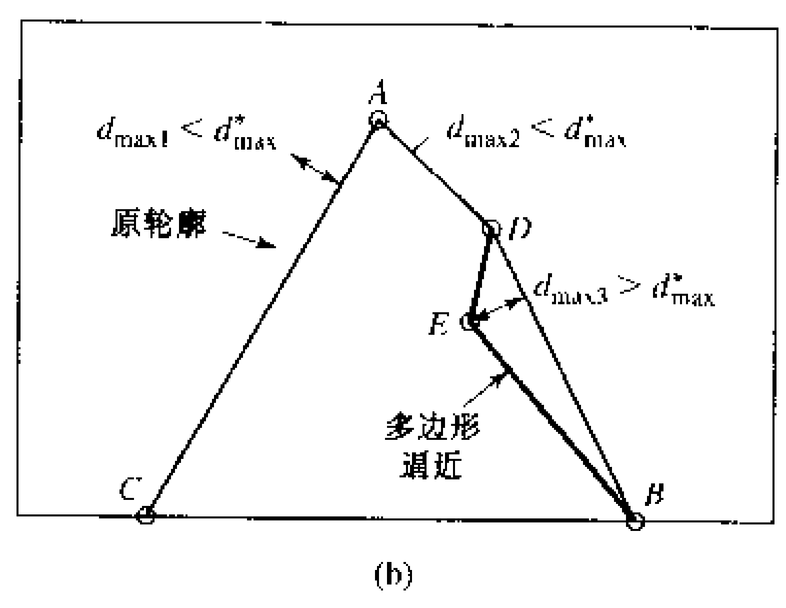


图 10.5 轮廓的多边形逼近。示出了质量测度 d^*_{max};在精细化轮廓描述期间,点 E 被加在具有最大 d^*_{max} 的位置处

由于多边形表示允许有损形状编码,我们必须定义一个质量测度,以便可以计算编码形状的质量。通常是采用原始轮廓与近似轮廓位置之间的最大欧几里得距离 d_{max}。

多边形逼近的效率不仅取决于物体形状,也取决于轮廓描述的起点位置。多边形逼近不

采用为所有起点计算这种表示并从中选择一个效率最高的起点的办法，而是通过用那些具有最大距离的两个轮廓点作为起点进行计算的。然后，在多边形与边界之间的逼近误差为最大的地方加入另外的点到多边形上。图 10.5(a)中，初始逼近$\overline{AB}$先被点 C 扩展，然后被点 D 扩展。重复这个过程直到形状逼近误差小于阈值 d^*_{max}。图 10.5(b)示出了点 E 被加在测量到最大误差 d_{max} 的逼近中。先进的算法选择在给定的比特预算下使逼近误差为最小的顶点，方法是把顶点最佳地放置在轮廓上或沿轮廓的一个窄条内[30,32]。

编码器必须把线段的顶点坐标传送给解码器。在编码了一个顶点的绝对位置后，后续顶点的坐标是被差分编码的，用图像尺寸作为顶点坐标的上限。

对于无损形状编码，多边形表示需要许多顶点。对于无损编码，链码通常优于多边形表示，而对于有损编码，多边形表示的效率更高。

样条逼近　样条函数是由选择的控制点定义的。我们可以用样条函数由离散点计算连续曲线。我们把连续边界坐标表示为轮廓长度 t 的函数，如 $\mathbf{c}(t)=[x(t),y(t)]$。可以用 n 个节点 $\mathbf{p}_n=[x_n,y_n]$按照下式逼近这个轮廓：

$$\mathbf{c}(t)=\sum_{n=0}^{N-1}\mathbf{p}_n\cdot B_{n,q}(t) \tag{10.1.4}$$

样条函数 $B_{n,q}$ 称为 B 样条，这里，字母表"B" 表示的是"基"(basis)。$B_{n,q}$ 取决于函数的阶 q 和轮廓上节点的位置 t_n。它是由以下的递归方程定义的：

$$B_{n,q}(t)=\frac{(t-t_n)\cdot B_{n,q-1}(t)}{t_{n+q-1}-t_n}+\frac{(t_{n+q}-t)\cdot B_{n+1,q-1}(t)}{t_{n+q}-t_{n-1}};\quad q>0 \tag{10.1.5}$$

和

$$B_{n,0}(t)=\begin{cases}1 & t_n\leqslant t<t_{n+1}\\ 0 & 其他\end{cases} \tag{10.1.6}$$

图 10.6 示出了对于相等间隔的节点不同样条函数的形状。节点也可以在轮廓上以任意间隔放置，并允许节点重合。q 阶样条函数的性质是它前面的 $q-1$ 阶导数在轮廓上是连续的。如所看到的，公式(10.1.6)是一个简单的保持函数。q 阶的样条函数是由 $q-1$ 阶样条函数与 0 阶样条函数的卷积生成的。由于样条函数的长度是有限的，点 $\mathbf{c}(t)$ 的位置受 $q+1$ 个点的影响，这里，q 是样条的阶数。经常使用的是二阶和三阶样条。对于形状编码，我们必须用 I 个轮廓点逼近离散轮廓 $\mathbf{c}(t_i)$：

$$\mathbf{c}(t_i)=\sum_{n=0}^{N-1}\mathbf{p}_n\cdot B_{n,q}(t_i),\quad 0\leqslant i<I \tag{10.1.7}$$

当选择节点数 N 小于轮廓点数 I 时，我们可以计算节点 $\mathbf{p}_n$。在这种情况下，我们用最小平方法解这个方程组。

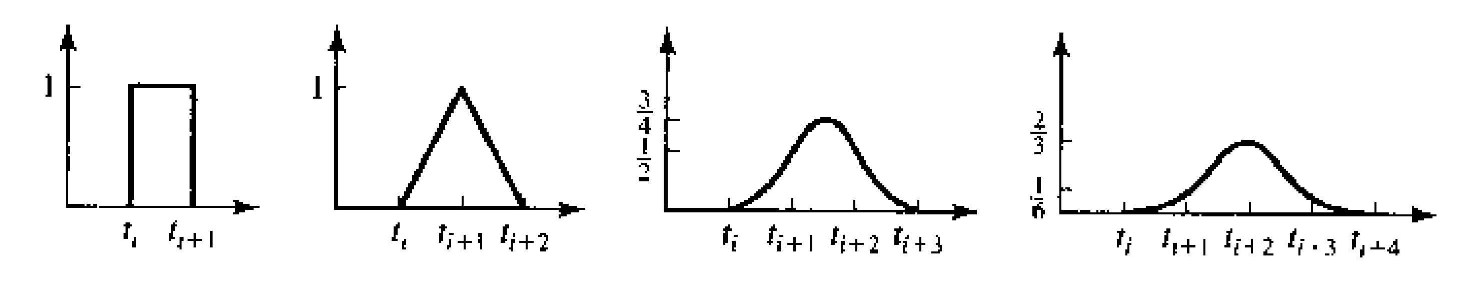


图 10.6　阶数 $q=0,1$(线性的)，2(二阶的)，3(三阶的)的具有等距节点的样条函数

在参考文献[25]中,用B样条曲线逼近边界。为了寻找控制点的最佳位置,通过使边界与逼近之间的MSE为最小来公式化最优过程。当平滑边界是主要问题时,这是一个合适的目标函数。然而,当所得的控制点需要编码时,必须考虑编码成本与所产生的失真之间的平衡。通过选择MSE作为失真测度并允许控制点位于平面的任何位置,所得到的最优问题是连续和凸出的,并且可以容易地求解。然而,为了高效地编码所产生的控制点的位置,必须量化它们,因此就会失去解的最优性。众所周知,离散最优化问题(被量化的位置)的最佳解不必接近相应的连续问题的解。在参考文献[30]中,给出了使用样条的形状编码器。编码在运算率失真的意义上是最佳的;也就是说,控制点的放置要求对于给定的失真使所需的比特率为最小。

傅里叶描述符 为了识别方面的应用,开发了傅里叶描述符;在这种应用场合形状是非常关键的。傅里叶描述符使封闭轮廓的平移不变、旋转不变和缩放不变表示成为可能[60]。有不同方法定义封闭轮廓的傅里叶描述符。用一个简单的描述符,我们在图像平面上顺时针地采样具有 N 个像素的轮廓上的二维坐标(x_n, y_n)。我们把坐标表示为复数 $z_n = x_n + j \cdot y_n$,并计算傅里叶级数为:

$$Z(m) = \sum_{n=0}^{N-1} z(n) \cdot e^{-j(2\pi nm/N)} \tag{10.1.8}$$

平滑轮廓的能量集中在傅里叶级数的低频。通过把高频系数置为零来平滑轮廓。

从公式(10.1.8)可以看出,轮廓的平移将只影响系数 $Z(0)$,即DC值。轮廓的缩放将使所有的系数乘以缩放因子。轮廓的旋转将平移傅里叶系数的相位但不影响它们的幅度。

另一种计算傅里叶描述符的方法是把二维坐标序列(x_n, y_n)变换成一个有序序列$(n, (y_{n+1} - y_n)/(x_{n+1} - x_n))$,同时$(x_0, y_0) = (x_N, y_N)$,并且 $0 \leqslant n \leqslant N$ 为轮廓点。项$(y_{n+1} - y_n)/(x_{n+1} - x_n)$描述轮廓方向的变化。这种描述符的傅里叶级数把更多的能量压缩到较低的系数。这一部分是由于DC值描述一个圆圈,具有恒定的方向变化,而不是在公式(10.1.8)中的直线。

为了保持形状的主要特征,只有大的傅里叶系数必须保留。傅里叶描述符在重建仅用几个系数的类似多边形的形状时效率不高。这就是它们在编码效率方面从来都不很具有竞争力的原因之一。

10.1.3 形状编码效率的评价准则

有损形状编码可以产生两类不同的失真:

- 物体的轮廓可能会改变:尖角被平滑或部分轮廓移动少量的像素。
- 物体的拓扑结构可能会改变:孔可能被添加或封闭,或者部分物体与原始形状断开。

通常用两种质量测度客观地评价编码的形状参数的质量。峰值偏差 d_{max}^* 是每个编码的轮廓点与原始轮廓上最接近的点之间的最大欧几里得距离。由于物体轮廓的变化,这种测度能够容易地说明形状质量。然而,如果有损形状编码造成了物体拓扑结构的改变,峰值偏差 d_{max}^* 就不是一个有用的测度。第二种测度 d_n 依赖于属于物体的像素数和它的编码表示。度量 d_n 是编码的形状被错误地表示的像素数除以原始形状的总像素数;即 d_n 描述相对的区域误差。由于不同物体可以具有很不同的轮廓像素与内部像素之比,对于 d_n,一个给定的值只允许我们比较同一个物体的不同逼近。质量测度 d_n 本身不提供足够的关于主观形状质量的信息,因

为一个给定 d_n 的主观含义不仅依赖于物体的大小也依赖于轮廓的长度。

几个序列的主观评价表明，具有逼近误差 $d^*_{max} > 3$ 像素的形状表示对视频是根本没用的，与它的分辨率无关。已经发现，在 CIF 分辨率（352 × 288 个像素）下，一个峰值距离 $d^*_{max} = 1.4$ 像素足以允许适当地表示低码率视频编码应用系统中的物体。

主观评价也表明，当比较不同的基于位图或基于轮廓的形状编码器时，上述两个客观测度真实地反映了主观质量。对于有损形状编码，基于位图的形状编码器产生块状的物体形状，而基于轮廓的编码器产生的物体形状将表现出曲线失真，尤其在物体的拐角处或多边形的边缘处。由于这两类形状编码器产生不同的失真（图 10.7），属于不同类的算法之间的比较必须主观地进行。

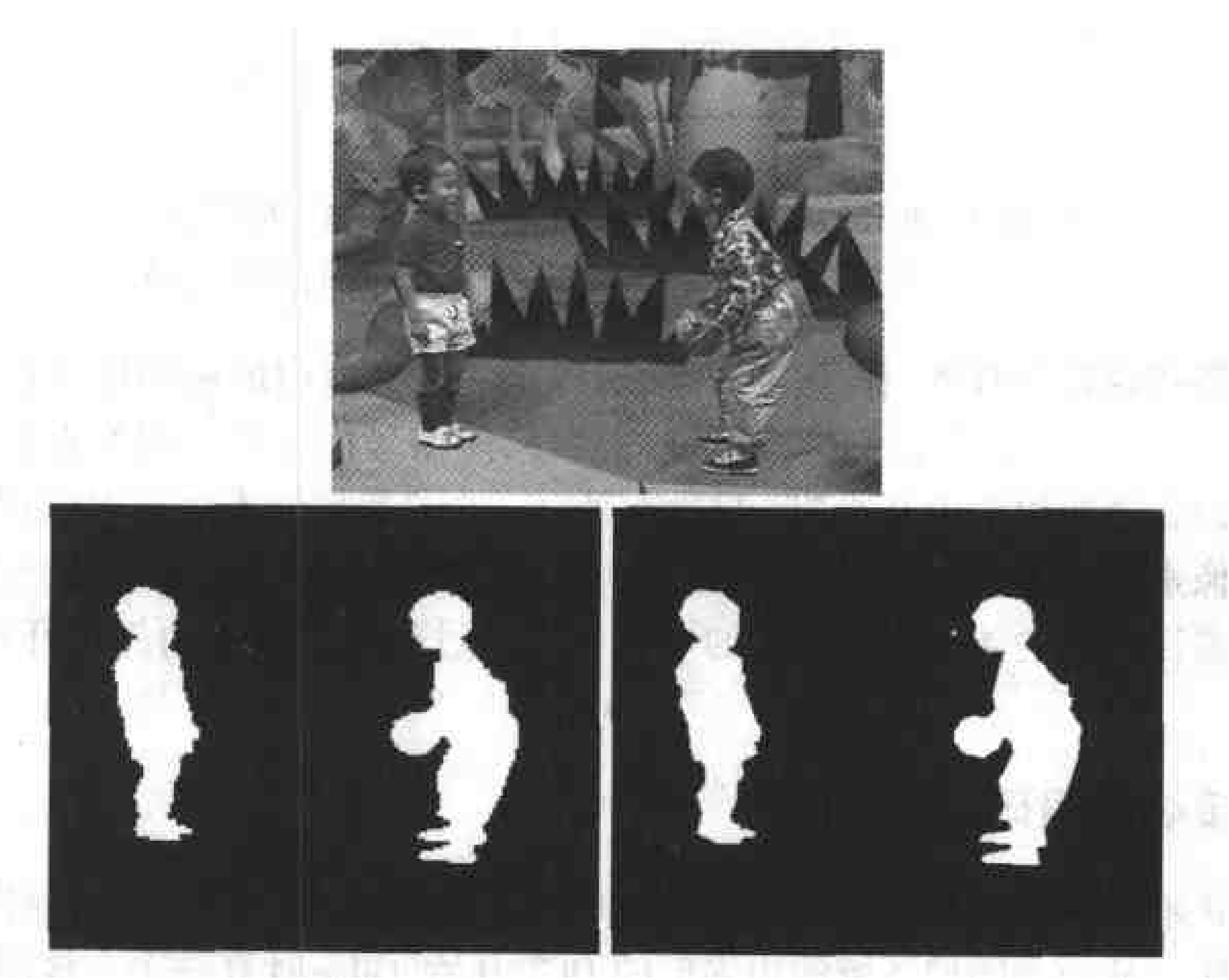


图 10.7　使用基于位图的（左）和基于轮廓的（右）形状编码器的有损编码。形状描述在上面图像中的两个儿童

10.2　对于任意形状区域的纹理编码

对于任意形状区域的纹理编码有两类算法：

1. 在第一类中，我们外推区域的纹理以填充一个矩形。然后，使用一个矩形区域纹理的编码算法编码这个矩形。在 10.2.1 节中给出了例子。这些算法一般是快速计算的，且会产生与矩形中的像素一样多的系数（与第二类算法比较，这是它的一个缺点）。
2. 在第二类中，直接用一个适应于区域形状的变换来编码该区域的像素。因此我们计算与区域中像素一样多的系数。在 10.2.2 节中将给出例子。

10.2.1　纹理外推

纹理外推或填充算法在区域周围用一个边界框来定义欲编码的纹理区域。我们称属于区域的像素为不透明的，而称在区域外但在边界框内的像素为透明的，因为它们对于解码器来说

是不可见的。边界框的纹理将用任何适于编码矩形区域纹理的算法进行编码，例如 DCT 或小波变换。图 10.8 示出了一个区域的边界框。这个边界框被加以扩展，以便使它的宽度和高度适于纹理编码算法。如果采用 DCT，我们可以扩展这个矩形，使它的宽度和高度为 8 或 16 个像素的倍数。

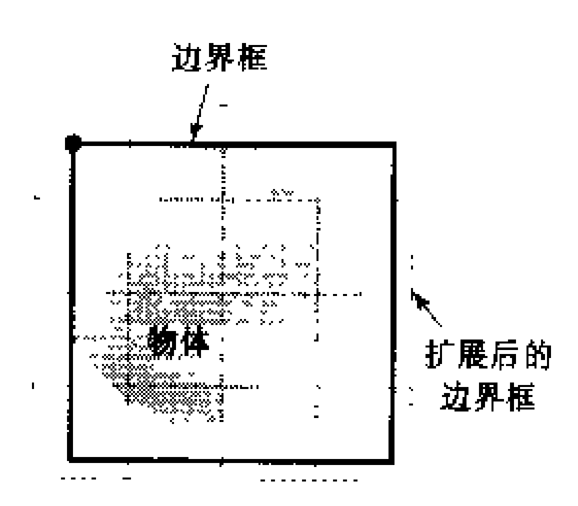


图 10.8 物体或感兴趣区域的外接矩形被扩展，使扩展后边界框的边长为8的倍数，以便进行DCT编码

由于解码器将忽略所有透明像素的纹理，所以定义边界框内的透明像素使它们不会无必要地增加码率。理想情况下，我们外推区域的纹理，使信号扩展到透明像素而不产生高频分量。这通常把设置透明像素为 0 或 255 排除在外。一个最简单的方法是把纹理置为该区域的或相邻不透明像素的纹理值的平均值。另外，可以把低通外推滤波器应用于这些透明像素。把每个透明像素设为它的 4 个相邻像素的平均。这个过程从边界框的一角开始，并以扫描行顺序进行。

10.2.2 直接纹理编码

为了把欲传输的系数的数目限制为必须编码的像素数目，人们开发了几种计算任意图像区域变换的算法。从一个图像区域的边界框的 DCT 开始，Gilge 计算一个在该图像块上为正交的变换[16]。所得到的变换计算出与图像块中像素个数一样多的系数。而且，它利用了相邻像素之间的空间相关性。对于图块的每个新的形状，必须计算一个新的变换，使得这种方法的计算量特别大。

形状自适应 DCT(SA-DCT) 采用一维 DCT 变换，计算起来快得多。首先，我们把一个图像块的所有像素垂直地移到块边界(图 10.9(b)) [55]。然后按照每列的长度应用一维 DCT。在第二步中，计算出来的 DCT 系数被水平地移到块的左边界（图 10.9(c)）。依照行中 DCT 系数的个数的长度再一次应用一维 DCT。最后，以与常规二维 DCT 系数一样的方式对这些系数进行量化和编码。由于 DCT 长度不同，SA-DCT 不是正交的。而且，移动像素不允许我们充分利用相邻像素之间的空间相关性。因此，SA - DCT 不如 Gilge 变换有效。与填充比较，这个比较复杂的算法的增益，对于相同的比特率在一个图像块上测得的 PSNR 平均为 1 ~ 3 dB。

小波编码器也可以适用于任意形状的区域[36]。为了计算小波系数，图块边界上的图像信号必须周期地且对称地扩展。由于没有像素被移动，因此保留了相邻像素之间的空间相关性。

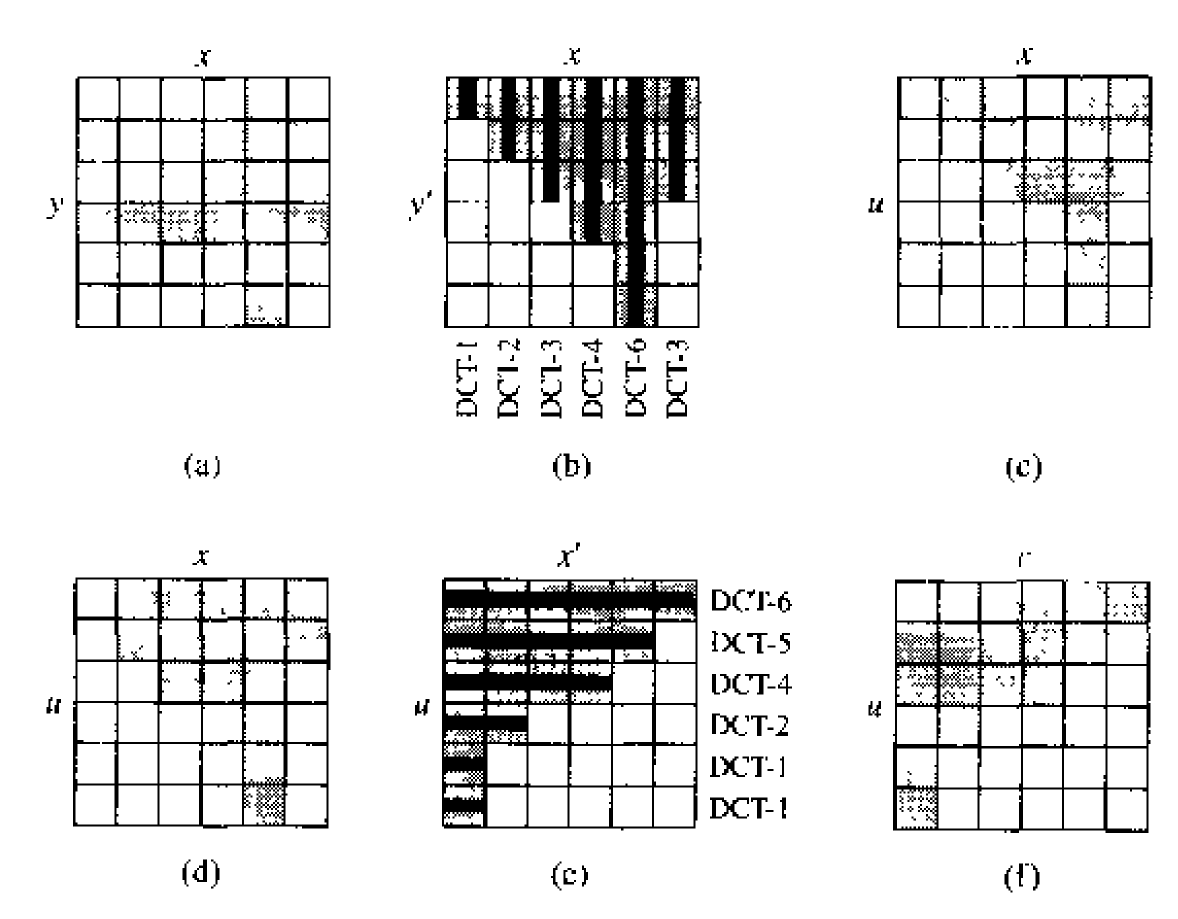


图10.9 一个形状自适应的DCT需要长度为 n 的变换：(a)原始图像块；(b)垂直移动的像素；(c)在垂直一维DCT之后DCT系数的位置；(d)在水平一维DCT之前DCT系数的位置；(e)水平移动的DCT系数；(f)在水平DCT之后DCT系数的位置

10.3 形状与纹理联合编码

这种形状编码技术是受到电影和TV演播室所用的蓝屏技术的启发而产生的。欲编码的物体放在静止的单色背景上。背景彩色必须在物体纹理所占据的彩色空间之外。通常，高饱和度的彩色(例如纯蓝)满足这一要求。由于形状信息作为纹理信息的一部分进行编码，我们有时称这种编码方法为隐含的形状编码。我们给出利用这一特性的两种编码算法。

目前，GIF89a[17,4]被用于环球网应用系统中，允许描述任意形状的图像物体。具有最多256色的图像基于一个称为LZW(由作者Lempel，Ziv和Welch而得名)的无损压缩方案进行编码。这种特殊的实现称为LZ78。在GIF文件头中，可以把这256种彩色之一声明为透明色。所有具有这种彩色的像素都不会显示出来，因此它们看起来是透明的。

这种定义透明彩色的概念也被引入视频编码。用基于帧的编码器编码视频信号。色度键(背景色)被传送到解码器。解码器对图像进行解码。与色度键具有相似彩色的像素被认为是透明的。否则，像素属于物体[6,22,20]。由于形状信息通常是由亚采样色度信号携带的，这种技术不适合无损形状编码。因为形状信息是被嵌入在纹理中的，所以只要存在纹理量化，这种形状编码就是有损的。这种方法的一个最重要的优点是它的低计算和算法复杂性。只要考虑编码效率，这种隐含的形状编码器就比直接的形状编码加纹理编码(例如SA-DCT或具有DCT的低通外推)需要更高的码率。

10.4 基于区域的视频编码

大多数图像和视频编码器都为了编码效率而做了优化。图像和视频的质量是由PSNR测量的。已经证明简单的PSNR测度不能很好地记录人类视觉系统的特性。在低比特率下这变

得非常明显，此时块失真使图像变形，但并不造成低的 PSNR。基于区域的图像和视频编码，也就是所谓的第二代图像和视频编码[34]，试图给予 HVS 以特别的关注。HVS 的性质导致以下的基本要求，这些要求为设计这些算法时做出选择奠定了基础[59]：

- 边缘和轮廓信息对于人类视觉系统是非常重要的，是我们的感觉所依赖的。
- 纹理信息具有相对的重要性，当与轮廓信息在一起时会影响我们的感觉。

在这些假设的基础上，基于区域的视频编码更注重编码轮廓而不是编码纹理。当确定重要的轮廓时，运动（即静止的或运动的物体）是不考虑的。

基于区域的视频编码器把每个图像分割为相似纹理的区域。由于轮廓被认为是非常重要，编码器以高精度传送区域的轮廓。区域的纹理是用原始区域的纹理的平均值近似的。图 10.10 示出一个被分割为区域的图像。不同的分割结果取决于相似性准则定义的精确程度和最小区域尺寸。与低码率下的基于 DCT 的图像编码器比较，这种编码器不产生任何块效应。然而，平坦的纹理表示会产生一类某些人认为更讨厌的新的失真。在高码率下，基于 DCT 的编码明显优于基于区域的编码，因为基于区域的编码需要传输许多轮廓。

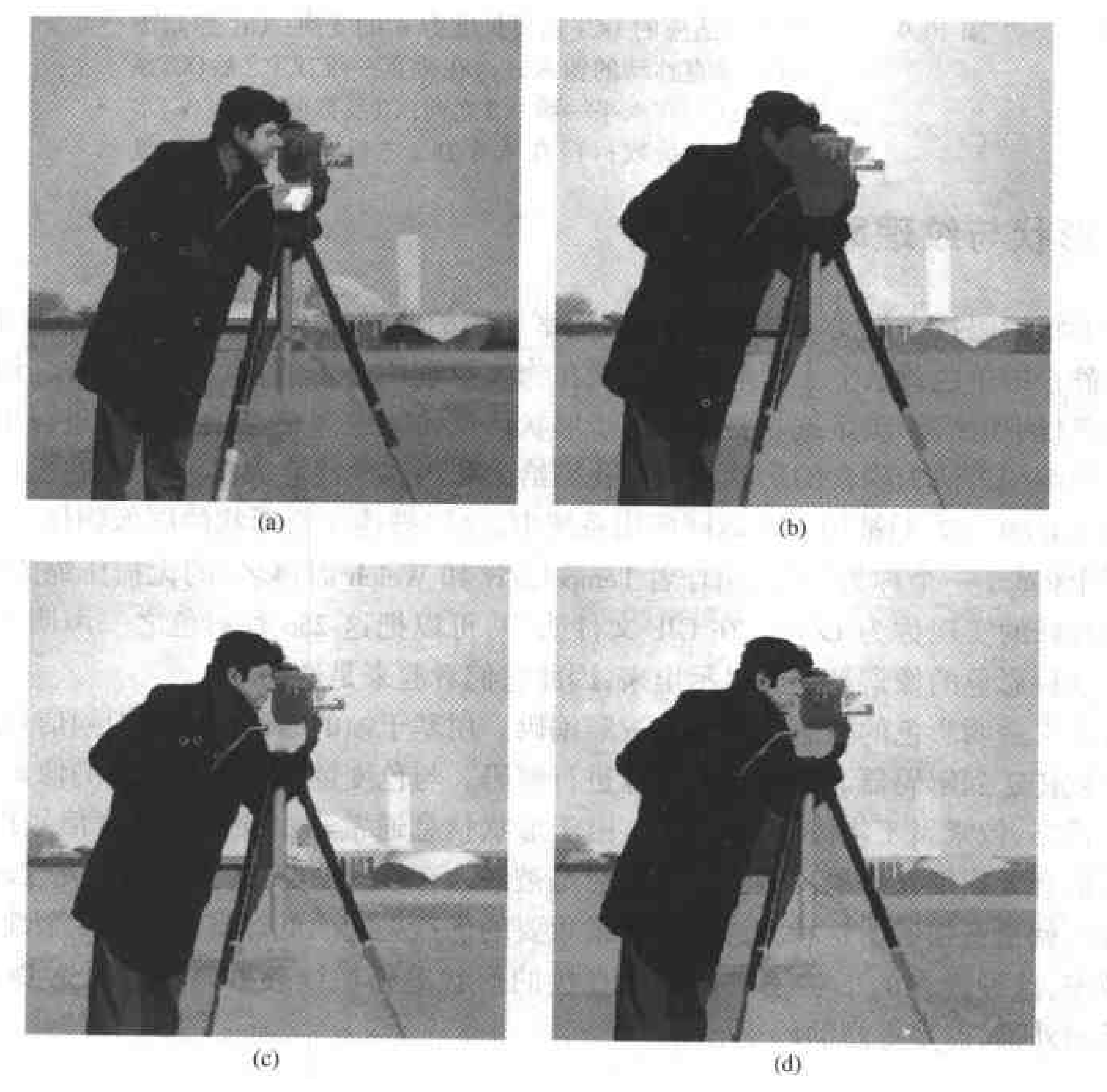


图 10.10　图像“摄影的人”被分割成几个区域。每个区域由它的轮廓和原始纹理的平均值表示。改变分割算法的参数可以实现不同分割（需要不同的编码数据量）

可以把这种概念扩展到视频编码[53]。为了减少形状编码所需要的码率，要从一幅图像到另一幅图像进行区域跟踪。把具有类似的或相同运动的区域聚合在一起。对于这个新的图像，

编码器传送当前图像中区域的运动、它们形状的变化以及新出现的区域。也传送纹理值的变化。

10.5 基于物体的视频编码

基于物体的分析合成编码(OBASC)[39]按照二维和三维运动模型定义均匀运动的物体。与基于区域的编码比较,OBASC 不进一步把物体分割成同类纹理的区域,从而省去了形状参数的传输。OBASC 把序列中的每个图像划分成均匀运动的物体,并以分别定义物体的运动、形状和彩色的三个参数集 $A(m)$,$M(m)$和 $S(m)$描述每个物体 m。运动参数定义物体的位置和运动。彩色参数表示物体表面的亮度以及色度反射率(在计算机图形学中,这些有时称为纹理)。图 10.11 解释了 OBASC 的概念和结构。OBASC 不需要如在基于块的混合编码中所用的帧存储,而需要一个存储器存储欲编码和传输的物体参数 $A'(m)$,$M'(m)$和 $S'(m)$。在编码器和解码器中的参数存储器包含相同的信息。通过估计这些参数集,图像合成计算显示在解码器的模型图像ψ'_k。存储器中的参数集和当前图像ψ_{k+1}是图像分析的输入。

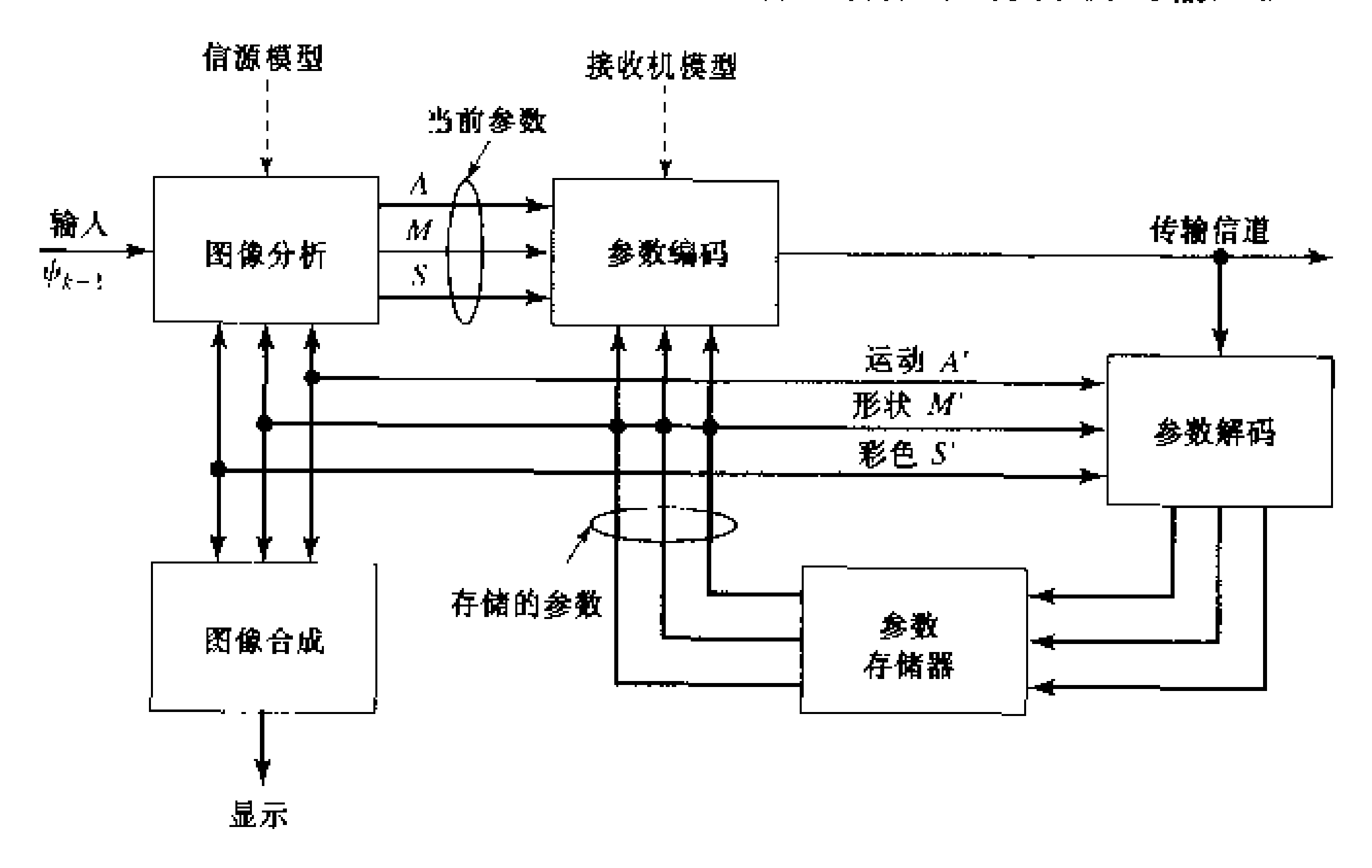


图 10.11　基于物体的分析合成编码器的方框图

图像分析的任务是分析欲编码的当前图像ψ_{k+1}和估计每个物体 m 的参数集 $A_{k+1}(m)$,$M_{k+1}(m)$和 $S_{k+1}(m)$。在当前图像中,首先检测运动的和静止的物体。对于运动物体,估计新的运动和形状参数,而重新利用大多数已传输的彩色参数 $S'_k(m)$。运动和形状参数能够正确估计出来的物体记为 MC(模型一致性)物体。在图像分析的最后一步,删除不能由 MC 物体用以前传输的彩色参数以及新的运动参数 $A_{k+1}(m)$和形状参数 $M_{k+1}(m)$描述的图像区域。这些模型失效(MF)的区域只由二维形状和彩色参数定义,并被称为 MF 物体。利用模型物体小的位置和形状误差(称为几何失真)不影响主观图像质量的事实进行 MF 物体的检测。这种假设对 OBASC 是成立的,因为基于物体的图像表示,运动补偿预测图像看起来像真实的图像。然而这种图像可能从语义上是不正确的。这样,MF 物体就简化成那些在运动补偿和形状补偿的预测图像与当前图像ψ_{k+1}之间有显著差异的图像区域。它们的尺寸偏小。这就允许我们以高质量编码 MF 物体的彩色参数,从而避免主观上的量化误差。由于从码率来说传输彩色参数是花费大的,MF 物体的总面积不应该超过图像面积的 4%,假设码率为 64 kbps,格式为 CIF,帧率为 10 Hz。

取决于物体类型是 MC 还是 MF，每个物体的参数集是由采用预测编码技术的参数编码进行编码的(图 10.12)。对于 MC 物体，运动和形状参数被编码、传输和解码；而对 MF 物体，形状和彩色参数被编码、传输和解码。因为就比特率而言编码彩色参数是昂贵的，所以必须联合设计参数编码和图像分析。设 R_A，R_M 和 R_S 分别表示运动、形状和纹理码率。通过使如下的总比特率为最小：

$$R = R_A + R_M + R_S \tag{10.5.1}$$

可以得到比基于块的混合编码技术更高的编码增益。

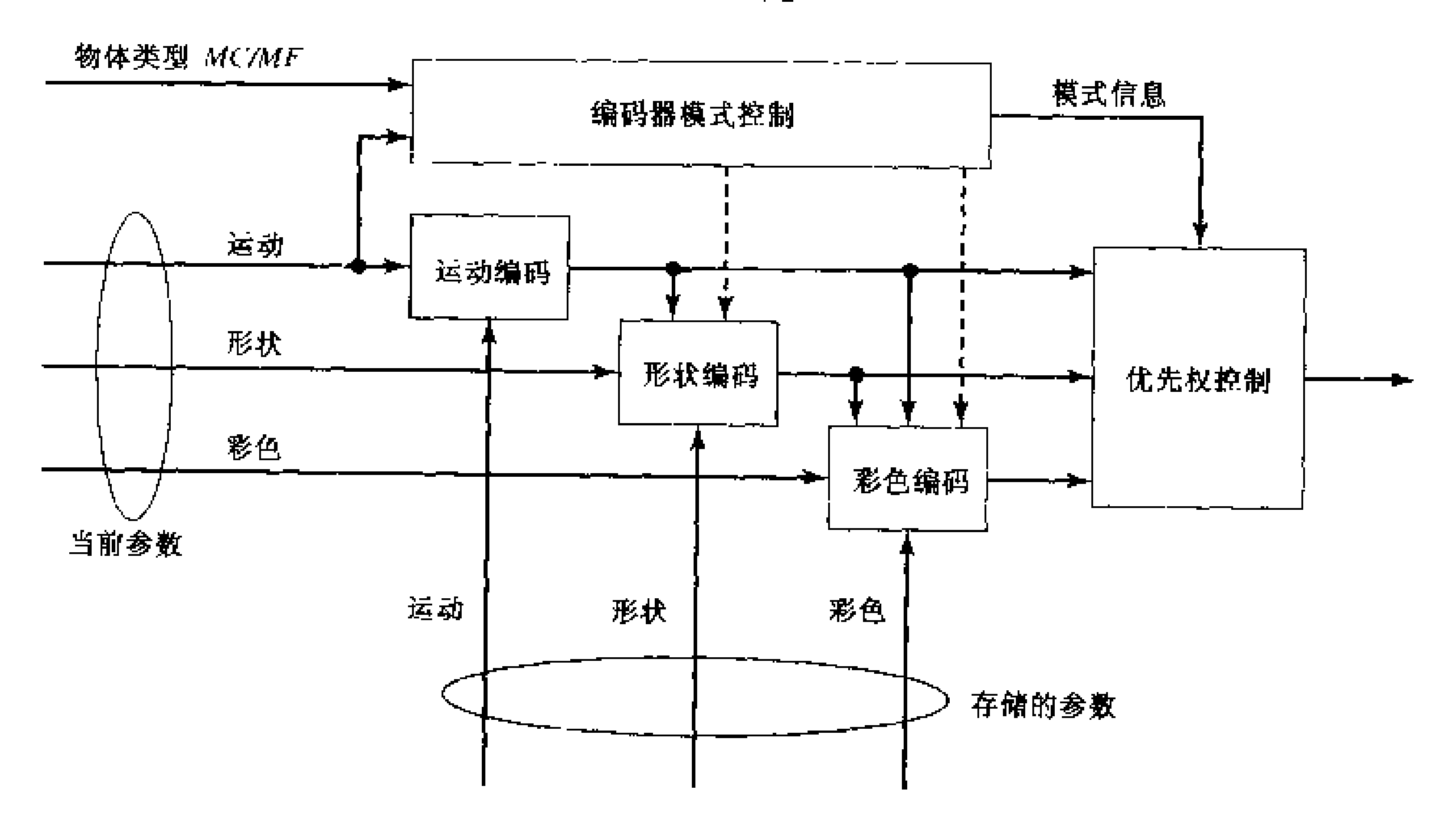


图 10.12 OBASC 的参数编码

在 OBASC 中，可以通过比较在相同的图像质量下编码同一个图像序列所需要的比特率来判断信源模型的适应性。图像质量主要是受检测模型失效的算法和编码模型失效的彩色参数所用的比特率影响的。以下我们描述几种基于 OBASC 发展起来的信源模型。

10.5.1 信源模型 F2D

柔性二维物体(F2D)信源模型假设真实物体的运动可以由平滑的位移矢量场描述。这个位移矢量场把真实物体在图像平面上的投影移到它的新位置。矢量场为每个属于物体在图像平面上的投影的像素定义一个矢量。由勾画物体在图像平面上的投影的二维轮廓定义物体的形状。

为了有效地描述物体的位移矢量场，对这个场进行亚采样，每 16×16 个像素仅传输一个矢量。解码器通过双线性内插重建位移矢量场，所产生的平滑位移矢量场使我们能够描述二维物体运动以及物体的某些柔性变形。例如，信源模型 F2D 非常适于模拟在平坦表面上运动的一片橡胶。另外，也可以描述橡胶的拉伸。运动物体的形状是用多边形逼近描述的。

基于块的混合编码使用运动补偿预测，由第 k 帧的纹理预测当前的图像 $k+1$，而 OBASC 把物体的纹理存储在纹理存储器中(图 10.13)。用这种纹理存储器改善运动补偿的性能。图 10.14 示出了依靠滤波器级联的运动补偿，与基于块的混合编码所做的一样。图像 k 的运动补偿预测，是通过在整像素运动矢量情况下从图像 $k-1$ 复制一个像素，在具有小数像素精度运动矢量的情况下用 $h(n)$ 进行滤波来计算的。低通滤波器 $h(n)$ 一般是作为双线性内插实现的。如果我们现在用帧 k 预测帧 $k+1$，可以看到，在图 10.14 中帧 $k+1$ 的像素 y_1 是通过

对前面已滤波的帧 k 的像素进行滤波来计算的。因此，当我们预测后面的帧时，用滤波器级联进行运动补偿预测会产生损失锐度的图像。

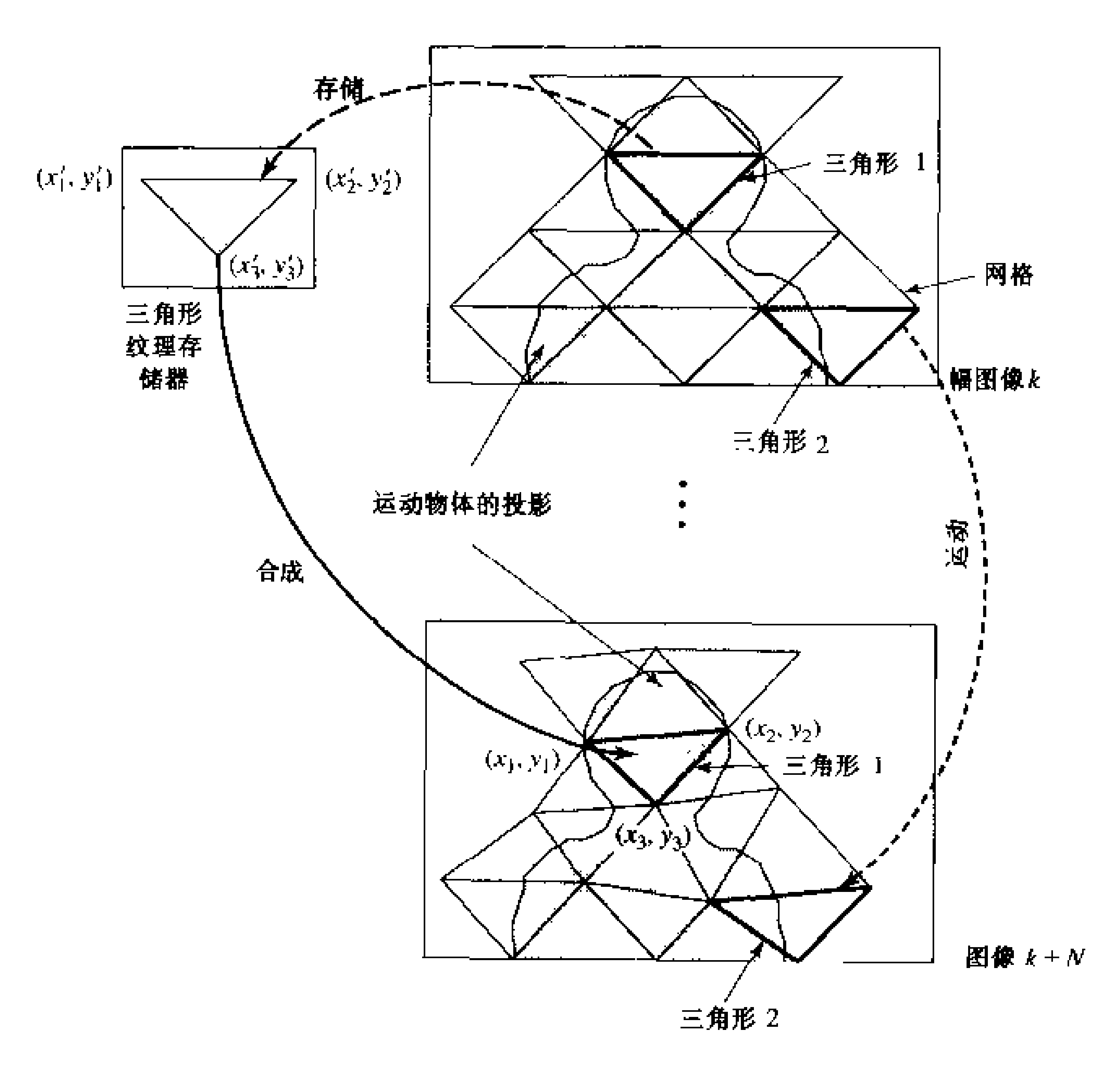
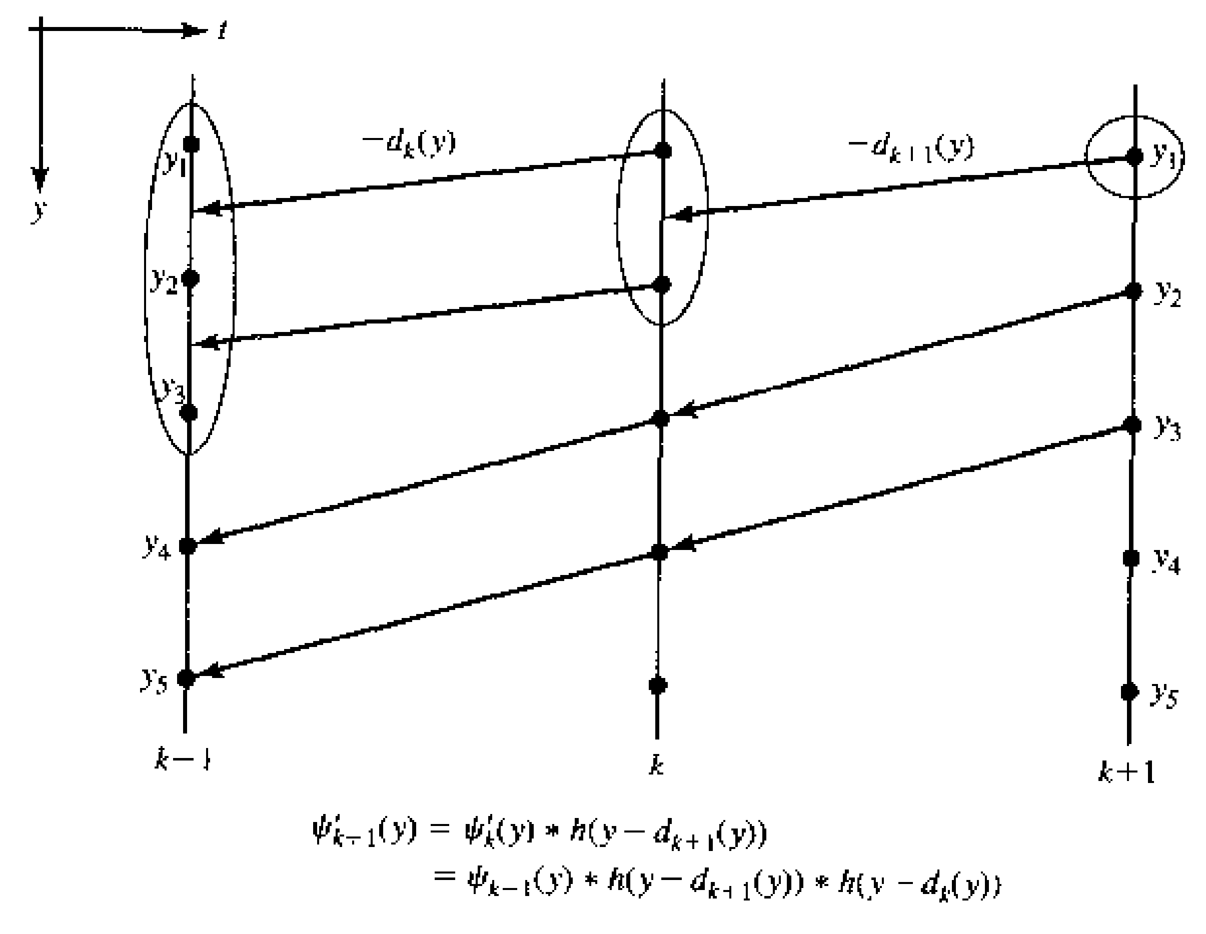


图 10.13　用基于三角形的网格作为纹理存储器的 MC 物体的图像合成



$$\psi'_{k+1}(y) = \psi'_k(y) * h(y - d_{k+1}(y))$$
$$= \psi_{k-1}(y) * h(y - d_{k+1}(y)) * h(y - d_k(y))$$

图 10.14　基于块的混合编码器用运动补偿滤波器的级联实现图像合成

用纹理存储器可以避免这种滤波器级联的效应(图 10.15)。再一次假设,这个纹理存储器对于帧 k 是用帧 k 的纹理初始化的,我们使用从帧 $k-1$ 到帧 k 的位移矢量场 $D_k(y)$,以便计算运动补偿预测图像 k。这要求对于一个小数像素的运动矢量,每像素进行一次滤波器运算。类似地,我们可以用级联的位移矢量场 $D_{k+1,tot}(y)$,通过直接访问纹理存储器,计算运动补偿预测图像 $k+1$。再一次,对于一个小数像素的运动矢量,我们仅需要每像素进行一次滤波器运算(见图 10.15)。

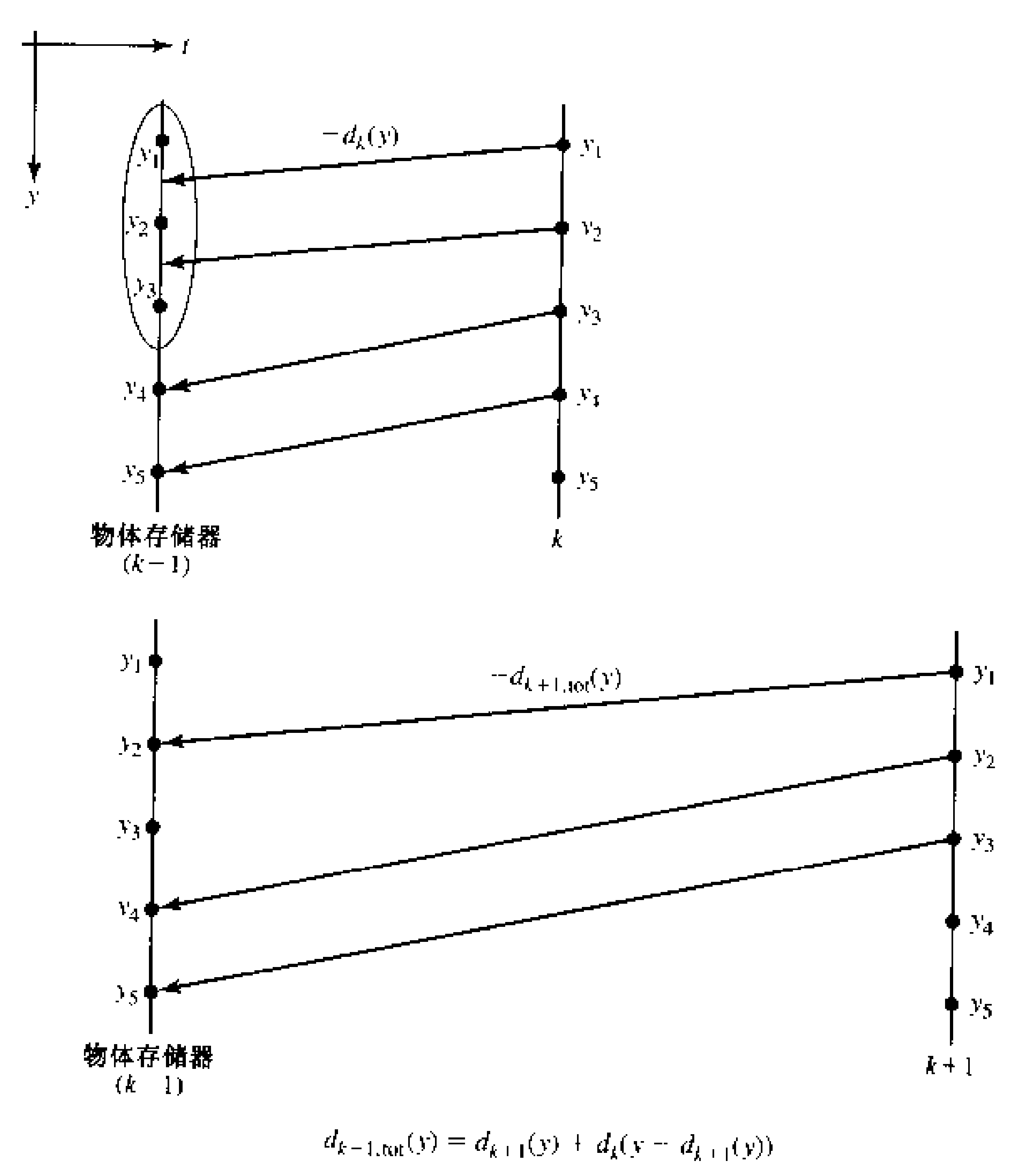


图 10.15　使用纹理存储器通过位移矢量场的级联进行图像合成(一维情况)

10.5.2　信源模型 R3D 和 F3D

使用具有三维物体模型的三维场景模型,我们可以用运动的刚性三维模型物体(R3D)更有效地描述和模拟真实世界的物体(5.3 节)。我们用线框描述模型物体的形状。用图像或纹理图描述物体的纹理。物体可以用公式(5.3.9)的三维运动形式在三维空间运动。为了能模拟接合处,我们允许物体由几个连接的、刚性的部分组成[37]。在运动和形状估计和补偿后,检测模型失效区域。预测图像与原始图像之间的差图像(图 10.16(a))用一个噪声自适应阈值进行二值化(图 10.16(b))。产生的掩膜确定那些在图像之间有差别的区域。某些差别是由运动补偿中的小误差引起的(图 10.16(c))。我们认为这些误差是几何失真,不具有主观上的重要性。重要的是模型失效区域,在那里图像的内容改变了。这些区域超出了最小尺寸,并

具有非线结构的形状。图 10.16(d) 示出了由眨眼、嘴运动和耳饰的镜面反射引起的模型失效(MF)物体。我们对模型失效区域 (MF_{R3D}物体) 仅编码其形状和纹理。

由于模型物体的表面被描述为线框,使得物体的柔性变形很简单。运动的柔性三维物体(F3D)的信源模型允许线框的控制点与物体表面成切线运动[45]。我们用二维位移矢量描述这种变形。这些位移矢量使我们能对真实物体的局部变形进行补偿。描述整个模型物体的柔性变形可能需要高的码率。因此,我们把位移矢量的使用限制于 MF_{R3D}物体区域。

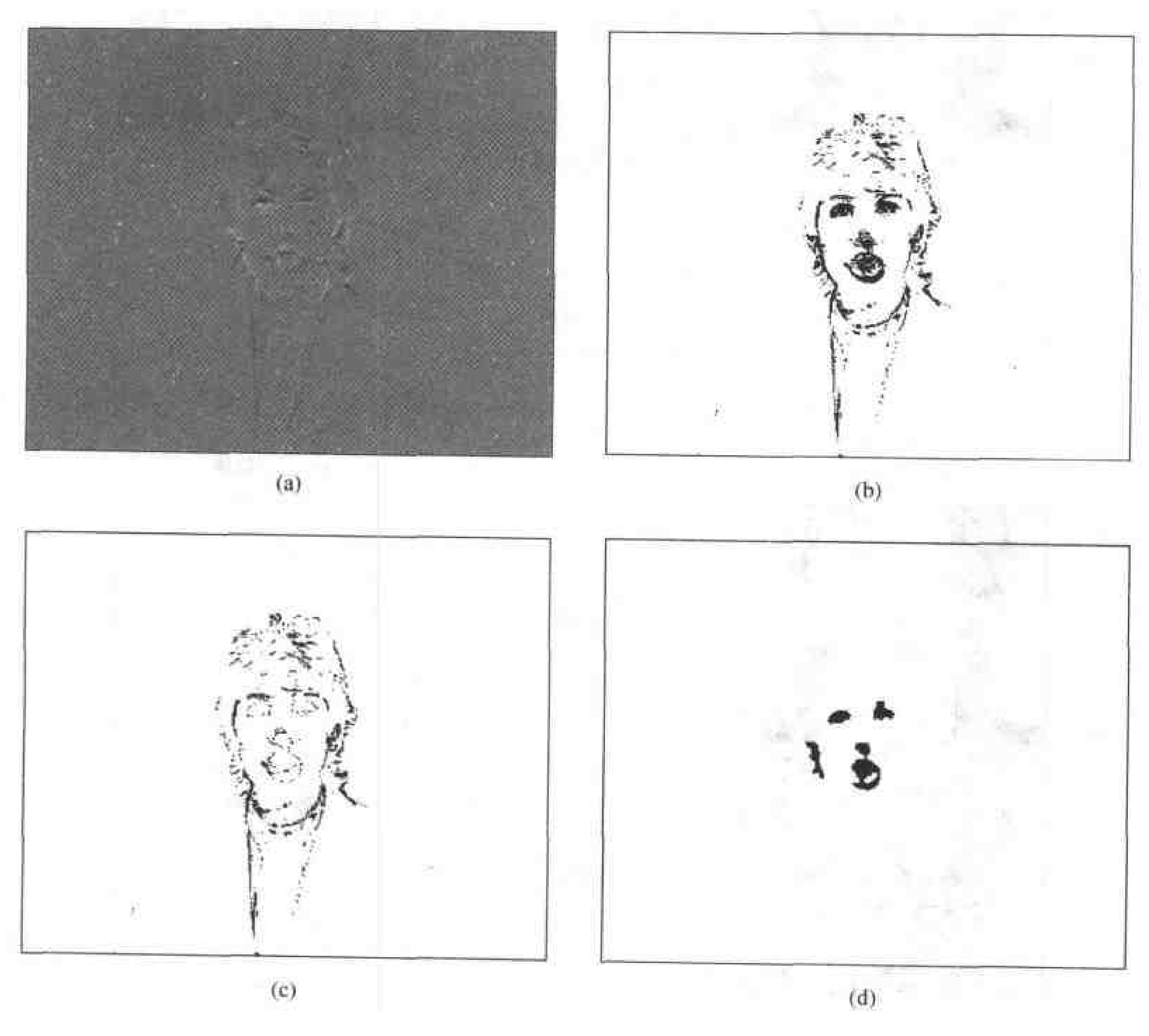


图 10.16　模型失效的检测:(a)缩放后的真实图像ψ_{k+1}与运动及形状补偿后的模型图像ψ_{k+1}之间的差图像;(b)综合误差掩膜;(c)几何失真和感觉上的不相关区域;(d) 掩膜 MF_{R3D}指示信源模型 R3D 的模型失效的区域

如在前面几段中所描述的,使用信源模型 F3D 要求在第一步中估计 MC_{R3D}物体和 MF_{R3D}物体。用于估计位移矢量的图像分析的输入包括 MC_{R3D}物体、MF_{R3D}物体以及真实图像 $k+1$(图 10.17)。为那些投影到当前 MF_{R3D}物体区域上的顶点估计位移矢量。对估计的位移矢量进行补偿后,我们用与信源模型 R3D 一样的算法,通过再次检测模型失效来核实这一步图像分析。图 10.17 示出了估计的位移矢量和 MF_{F3D}物体。这些 MF_{F3D}物体通常比 MF_{R3D}物体小,因为柔性位移矢量对一些柔性变形进行了补偿。由于编码 MC_{F3D}物体的位移矢量以及 MF_{F3D}物体的纹理和形状

花费较少的比特，因此不采用花费较大的 MF_{R3D} 物体的纹理和形状，采用 F3D 将使编码效率提高。这是可能的，因为如果它们减小了 MF 物体的尺寸，则码率仅花费在位移矢量上。当编码简单的视频电话序列时，把使用 R3D 换成使用 F3D，码率将从 64 kbps 降低到 56 kbps[46,47]。

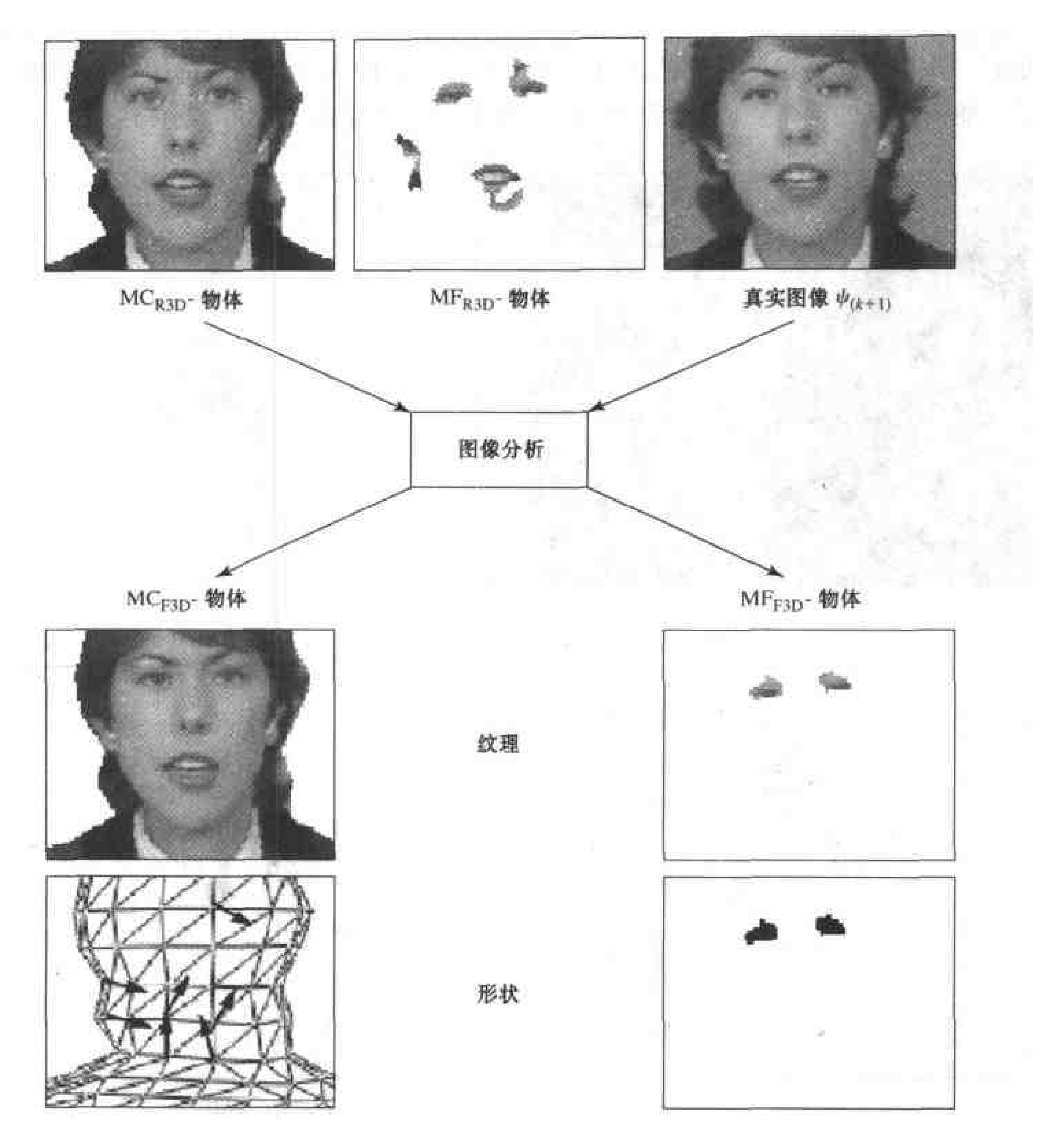


图 10.17 在 MF_{R3D} 物体区域估计柔性位移矢量。这通常引起 MF 物体尺寸的减小

OBASC 以每像素 1 比特的速率编码彩色参数，这比通常使用的基于块的混合编码器的每像素 0.1～0.3 比特高得多。因此，OBASC 表现出很少的量化效应，图像看起来比混合编码的图像更尖锐。然而，OBASC 对图像分析的依赖性强。必须正确地分割运动的物体，且必须相当精确地估计它们的运动。如果图像分析失败，OBASC 就不能用必须与运动和纹理参数一起传输的形状参数来增加编码效率。相反地，它可能需要比仅传输运动和纹理参数的混合编码器更高的码率。由于缺乏图像分析的健壮性，OBASC 仍是一个需要继续研究的领域。

取决于应用的信源模型，OBASC 把高达 80% 的码率用于对模型失效的编码上。OBASC 不需要利用任何场景知识。因此，检测模型失效的算法不受场景内容的控制。为了把关于场景的知识包括进来，我们必须把 OBASC 扩展到基于知识的分析合成编码（KBASC），它为有效地

编码而估计场景知识。

10.6 基于知识的视频编码

KBASC 编码器是基于 OBASC 的。另外，这种编码器试图识别视频场景中的物体（例如脸）。一旦编码器识别出物体，它就把物体的编码模式从一般的基于物体的模式切换到基于知识的模式。对于头和肩的场景，目的是要实现较好地模拟人脸，并利用人脸位置的知识控制编码器。为了能从 OBASC 自动切换到 KBASC，提出了一个识别脸特征以及图像序列的脸模型的自适应算法[28]。首先，我们必须确定想用 KBASC 编码的人眼和嘴（图 10.18(a)）。第二步，必须使脸模型适应当前描述人的线框（图 10.18(b)）。例如，我们把人脸模型"Candide"（图 10.18(c)）[49] 并入物体模型。我们可以水平地缩放脸模型以匹配两眼睛之间的距离，垂直地缩放脸模型以匹配嘴和眼之间的距离，从而使脸模型适应图像中人的脸。最后，当用 OBASC 层编码时，必须把脸模型缝合到用来描述物体的线框模型（图 10.18(d)）。

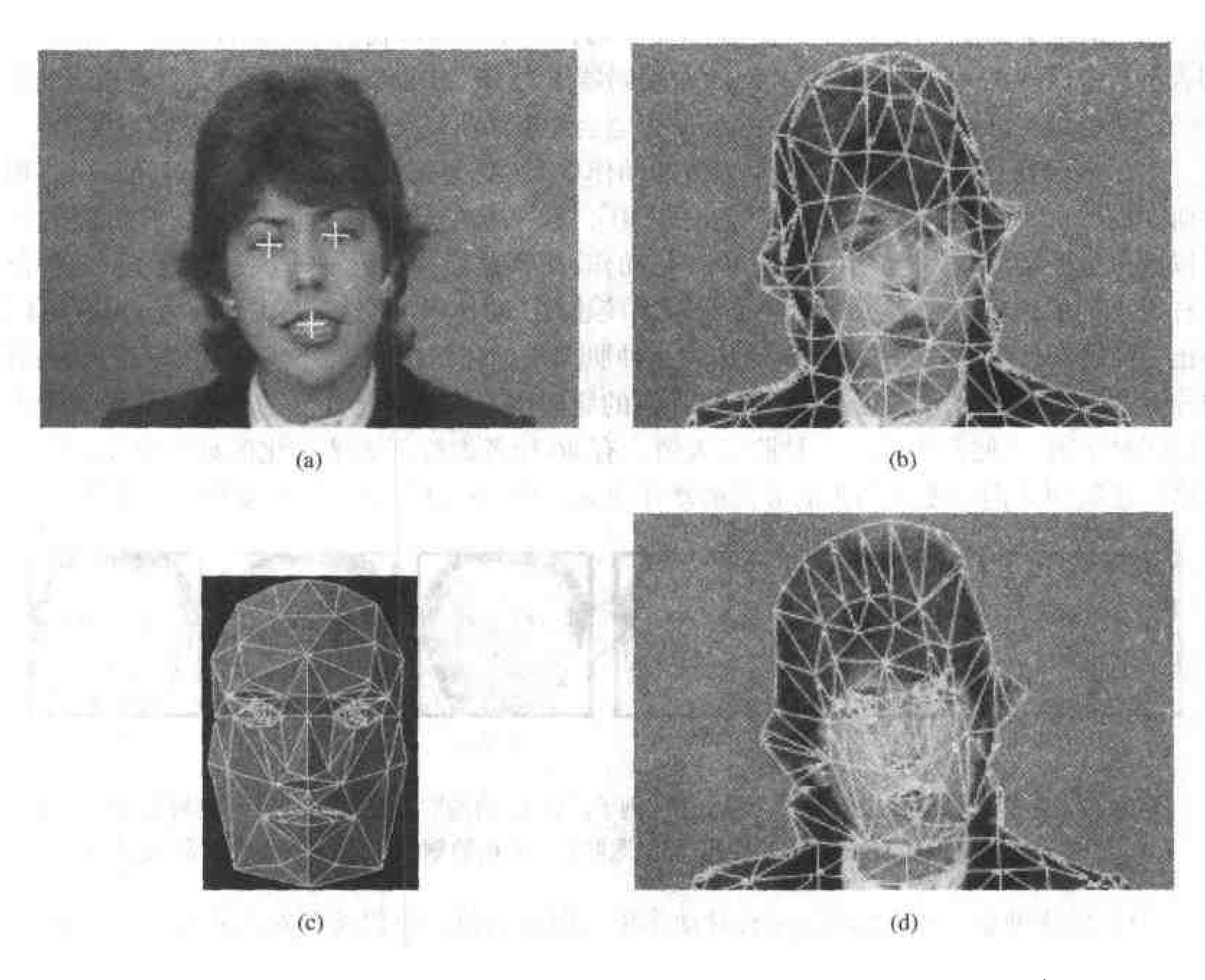


图 10.18　(a)首先，确定眼和嘴的位置；(b)用人的物体模型作为参考；(c)相对于这个图像缩放 脸模型；(d)最后，在线框的脸区域(b)切开一个开口并把脸模型缝合到该线框

一旦 KBASC 检测到脸，它的编码效率就会比 OBASC 有所提高。一个原因是，使用人脸模型而不是一般的模型改善了形状表示，并且允许更好的运动补偿。知道脸的位置也允许我们更精确地定义头和肩的接合处，这再次导致运动估计和补偿的改善。最后，我们可以利用关于

人脸位置的知识以基于场景的方式来计算模型失效。我们在脸部区域继续使用灵敏的检测算法,同时可以把脸部以外的大多数预测误差归因于几何失真以及照度误差,我们选择忽略这些误差。在一个空间分辨率为 352×288 像素、帧率为 10 Hz 的测试序列"Claire"的实验中,信源模型 F3D 的模型失效平均面积是 3%。当我们使人脸模型适应模型物体时,面积减少到 2.7%。最后,利用人脸位置的知识检测模型失效,把它们的面积减少到 2.5%,而主观图像质量没有受到损失。相应的码率对于采用 OBASC 编码器的 F3D 是 57 kbps;而对于 KBASC 是 47 kbps。

10.7 语义视频编码

前面描述的视频编码算法联合使用当前的图像信号和由时间预测导出的预测图像信号来描述视频序列的每一帧。语义视频编码用具有行为的模型物体描述视频序列,模型物体用真实物体的行为来代表真实物体。我们期望语义编码远比其他视频编码器有效,因为与为显示物体所需的像素阵列可能的变化相比较,大多数物体的运动和变形是非常有限的。视频表示中可能的变化数目决定了视频序列的最大熵以及所需要的比特率。如果我们用人脸模型考虑 KBASC,脸部表情从中性到喜悦的变化将要求 KBASC 编码器作为 MF 物体传输嘴的区域。具有适当的脸模型的语义编码器可以传输命令"微笑",而解码器会知道如何使人脸模型变形来使它微笑。

对于人脸模型,已提出描述脸部表情的动作单元(AU)[7,2,13,35]。动作单元是由心理学者研发的脸部动作编码系统(FACS)导出的[10]。该系统的开发是为了"确定所有可能的视觉可以区分的脸部运动"。所用的测量单元是动作,而不是肌肉单元——这是因为 FACS 联合了一种以上的肌肉单元来表示外观变化。该系统也用"动作单元"这一术语,因为 FACS 可以把不止一种动作归因于大多数解剖学家作为一种肌肉来描述的东西。例如,提动眉毛的额头肌肉被分为两个动作单元,取决于是内侧还是外侧的肌肉抬起内侧还是外侧的眼眉。几种动作单元可以同时运用,以便产生任意一种脸部表情。有 46 种考虑脸部表情变化的动作单元,有 12 种粗略描述凝视方向的变化和头的方位的动作单元。图 10.19 示出了几种动作单元的例子。



图 10.19 在参考文献[3]中使用的六种动作的例子。AU 1: 内侧眉毛抬高。AU 2: 外侧眉毛抬高。AU 4:眉毛降低。AU 5:上眼睑抬高(睁大眼睛)。AU 6:脸颊抬高。AU 7:眼睑收紧(部分斜视)

为了把这种系统用于脸部动画,对动作单元赋以幅值。可以按照对人进行测量的动作单元的开始、最高点和残余来模拟脸部表情暂时的行为。

FACS 系统是基于人类心理学的,而其他的建议定义脸上的特征点,使用脸部动画参数(FAP)使脸生动起来。在 MPEG-4 中使用的 68 个 FAP 是基于最小可察觉动作的研究,并且与肌肉动作密切相关[27,58,21]。FAP 表示了全部的基本脸部动作的集合,包括头运动、舌、眼以及嘴的控制。它们能够表示自然的脸部表情。图 10.20 示出了可以用来定义头形状的点以及可以活动的特征点。

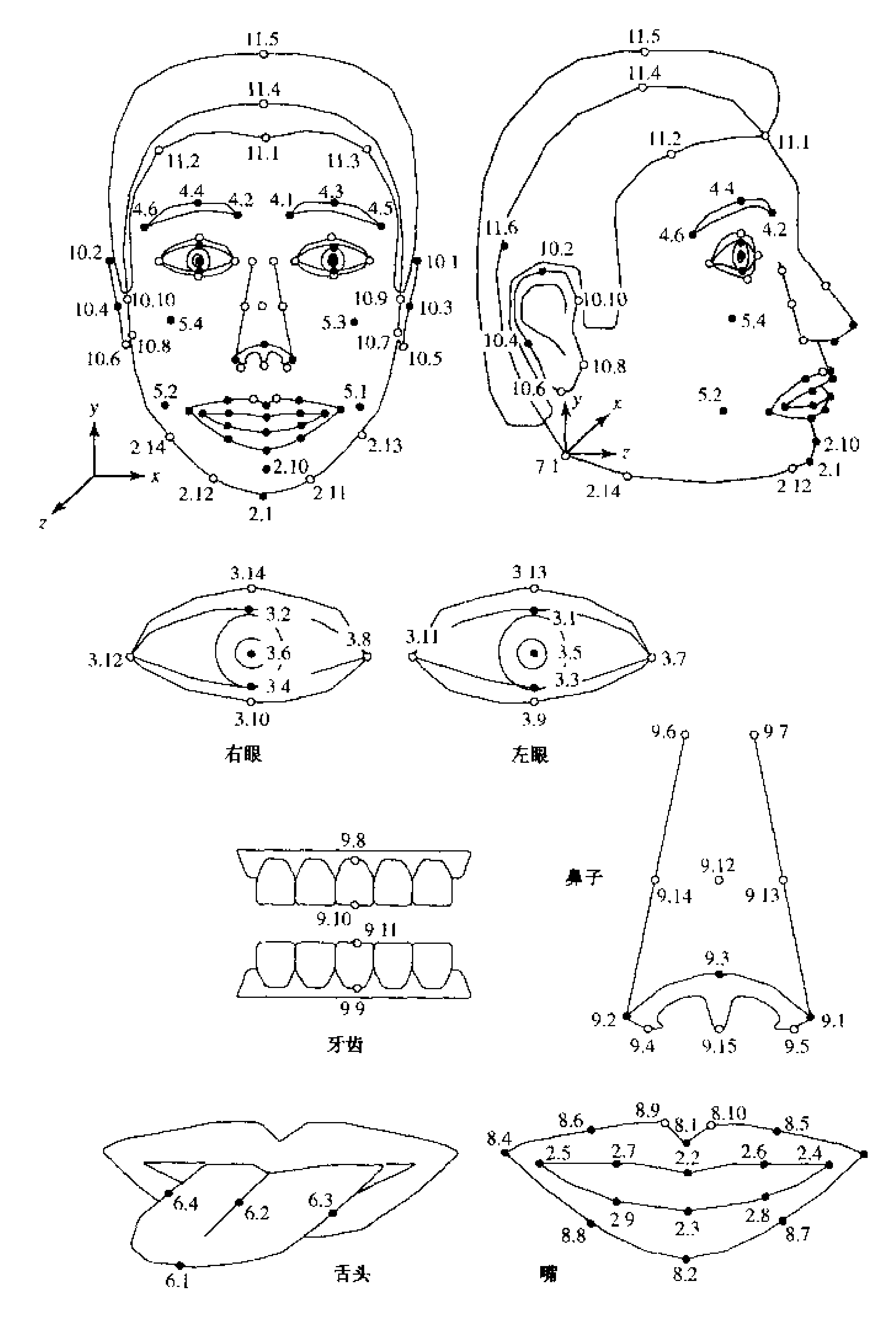


图 10.20　一般在 MPEG-4 中使用的脸的特征点

FAP 系统和 FACS 动画的脸具有不到 100 个参数。已经发现能够用不到 2 kbps 把 FAP 或动作单元用于脸的动画[57]。然而，还不清楚如何生成一个能够逼真地表现脸部活动的脸模型。形状、运动和照度估计仍是有待于解决的问题。而且，从视频序列中抽取像 FAP 或动作

单元那样的语义参数是一个困难的挑战。语义编码要求非常可靠的参数估计,因为估计的误差将在解码器中改变脸部表示的语义。高兴和悲伤两种表情的混合可能阻止有效的通信。如果用 KBASC 描述交通场景,一个误差就可能使汽车左拐而不是继续往前开。

由于语义编码器允许很高的编码效率,它将对通信信道中的误码非常敏感。比特差错将不会导致较低的图像质量,但会改变语义的内容。

10.8 分层编码系统

取决于场景的内容,在一个基于物体的编码器中用分离的物体模型描述每个运动物体可能不是高效的。类似地,以前面的图像作为参考来编码序列图像也不总是高效的。因此,我们需要用适当的信源模型集来编码视频序列。在参考文献[38]中,提出了分层编码系统(图 10.21)。这种系统允许应用不同的信源模型和不同水平的场景理解来编码图像序列,以便保证最好的性能。第Ⅰ层仅传输统计上依赖于像素的彩色参数,而第Ⅱ层允许附加传输具有固定大小和位置的块的运动参数。因此第Ⅰ层对应图像编码器或混合编码器的 I 帧编码器,而第Ⅱ层对应混合编码器的 P 帧编码。第Ⅲ层是分析合成编码器,它允许传输形状参数,以便描述任意形状的物体。这里,第Ⅲ层是用基于 F3D 信源模型的 OBASC 编码器表示的。

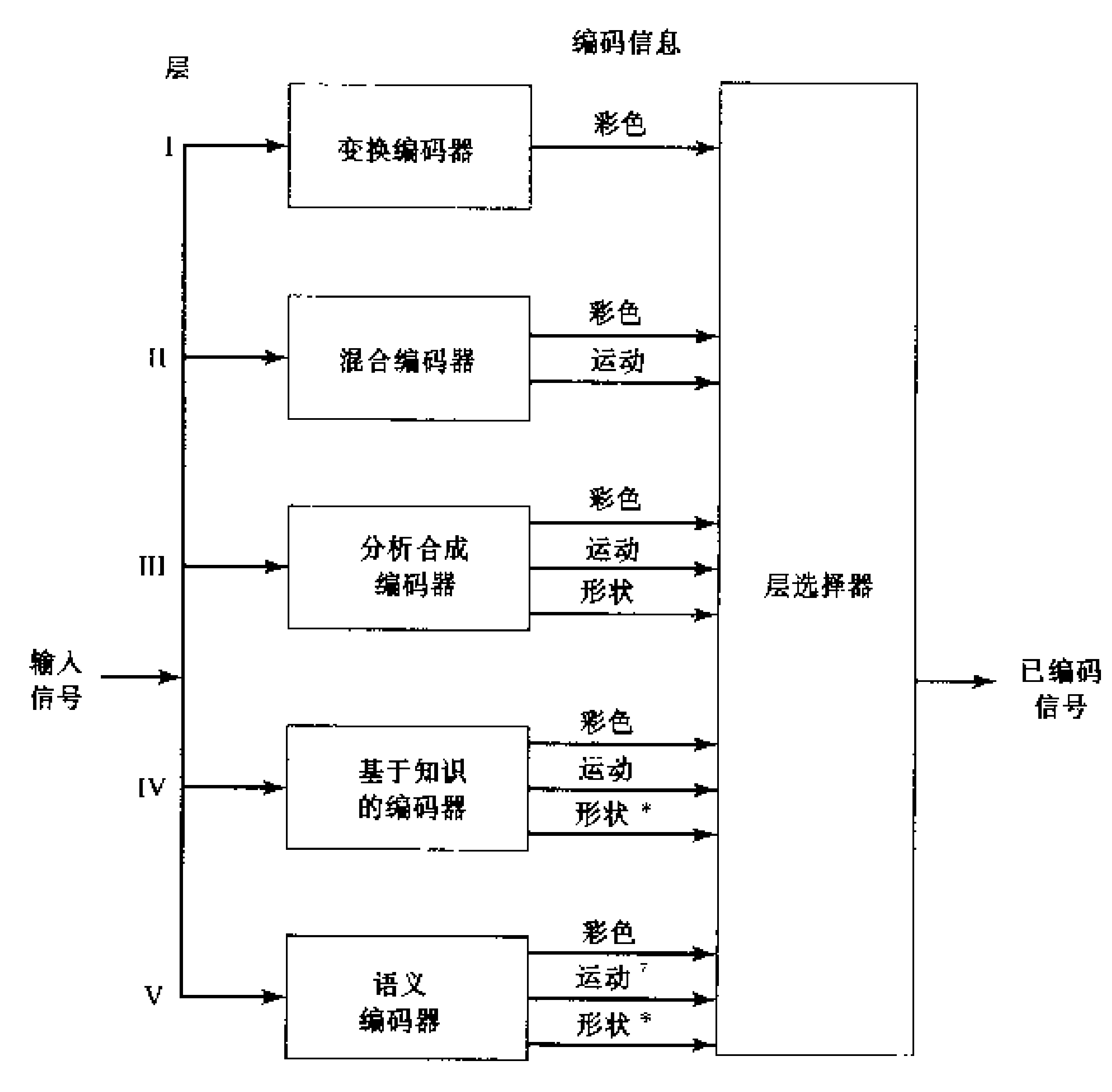


图 10.21 分层编码器的方框图

使用三维信源模型允许和 KBASC 的无缝合成。在第Ⅳ层中,为了应用基于场景的物体模型(例如,脸模型"Candide"),从视频序列中抽取并传输场景内容的知识。第Ⅴ层传输描述物体复杂行为的高水平符号。

这种分层编码系统的各个层已经由许多组进行了研究,然而选择适当的层进行编码仅有第Ⅰ层和第Ⅱ层被彻底研究过,因为这些层负责对基于块的混合编码器中的帧内模式或预测模式中的块进行编码。第Ⅱ层和第Ⅲ层之间的切换只是在最近进行了讨论。在参考文献[8]中,提出了一种根据残余编码误差和比特率的选择。在参考文献[44]中,运动物体的尺寸决定物体的编码是基于块的还是基于物体的层。在参考文献[28]中,讨论了在第Ⅲ层的 OBASC 和第Ⅳ层的 KBASC 之间的切换。在参考文献[12]中,研究了适于合成到分层编码系统中的语义编码器;与 H.263 视频编码器(第Ⅰ层和第Ⅱ层)比较,这种编码器在编码头肩视频序列时节省 35%的码率[22]。

10.9 小结

在本章中,我们讨论了基于内容的视频编码,它描述各个视频物体而不是整个帧。

- **二维形状编码(10.1 节)** 为了描述物体,我们用阿尔法映射定义帧中的每个像素是否属于物体。另外,阿尔法映射允许我们描述视频物体的任意的透明度。为了高效地传输视频物体的形状,我们考察了二维形状编码算法。基于位图的形状编码器用基于上下文的算术编码器描述每个像素是否属于物体,而基于轮廓的编码器用多边形、样条或链码描述物体的轮廓。人类视觉系统对于形状失真相对敏感。因此,我们倾向于以高质量编码形状。
- **纹理编码(10.2 节)** 涉及到纹理编码,我们现在应该能够编码任意形状区域的纹理。一种可能的方法是低通外推纹理使它充满一个方块,该方块能用如 DCT 之类的标准变换编码器编码。或者,我们可以用更有效的形状自适应 DCT 或小波变换,它们只产生与欲编码的像素一样多的系数。
- **基于物体的分析合成编码(10.5 节)** 基于物体的分析合成编码把任意形状的视频物体合成到视频编码的框架中。由于一个物体现在被描述为一个单元,运动补偿预测图像看起来像自然图像一样。它不会遭受基于块的运动补偿预测图像的块失真。我们用模型失效的概念来确定图像中那些不能以足够的主观质量模拟的部分。用二维和三维信源模型已经实现了 OBASC。三维信源模型被扩展到基于知识的编码,在那里编码器识别预先定义的模型,如人的脸,以便更有效地描述视频场景(10.6 节)。语义编码的目标是用高级描述来描述视频(10.7 节)。它已经为用于人脸进行了开发,其中如动作单元或 FAP 的参数集被用于脸的动画。
- **分层编码系统(10.8 节)** 最后,我们考察分层编码器,它把不同的编码模式合成到一个框架中。分层编码系统基于编码效率,从一个编码器切换到另一个编码器,保证我们总是用最有效的信源模型来编码视频场景。
- **物体分割和运动估计** 有效的基于内容的视频编码取决于精确描述图像内容的图像分析。图像分析包括分割物体或区域以及估计它们的运动的问题。分割涉及把初始帧划分成相关的纹理或运动区域,并一直跟踪这些区域。在 6.8 节中,我们简单地考虑了把帧分割成相关运动区域的问题。在那里我们看到,运动估计和区域分割是相互关联的,而且应该联合地进行最优化。分割和跟踪算法的更实质性讨论超出了本书的范围;感

兴趣的读者可以从参考文献[48,1,51,56]开始进一步研究这个课题。

10.10 习题

10.1 阿尔法映射是什么？在视频中如何使用？

10.2 两种主要的形状编码方法是什么？讨论它们的优点和缺点。当形状被有损编码时它们有何区别？

10.3 形状质量的客观测度是什么？

10.4 设想一条1个像素宽的对角线。当我们用具有4邻域或8邻域的像素定义一个物体时，这条线的表示如何变化？像素的六边形格栅的优点是什么？

10.5 对于图10.4中给出的例子，对于4邻域的直接链码和微分链码是什么？

10.6 对于图10.22中给出的形状，导出四叉树表示。

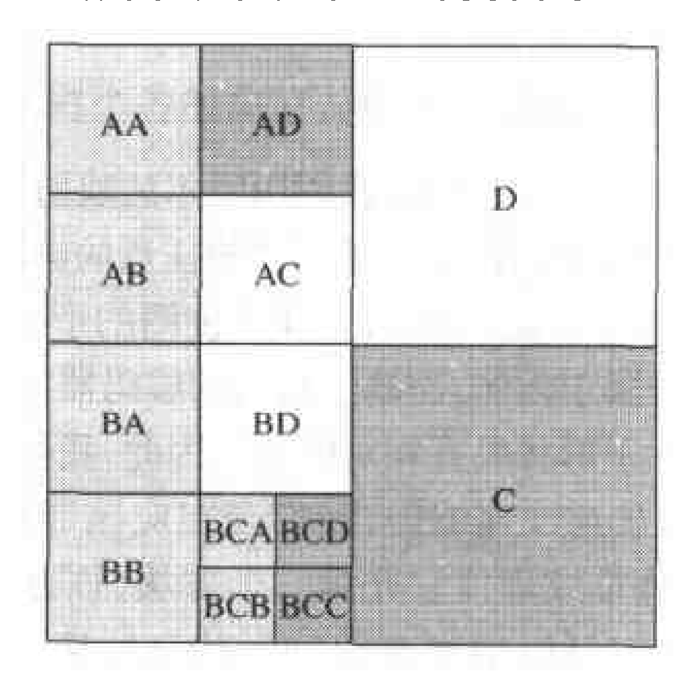


图10.22 具有三个用不同阴影表示的不同物体的图像

10.7 对于公式(10.1.8)给出的轮廓的傅里叶描述符表示，证明：(a)轮廓的位移将仅影响系数 $Z(0)$，即DC值；(b)轮廓的缩放将用一个缩放因子乘所有的系数；(c)轮廓的旋转将移动傅里叶系数的相位，但不会影响它们的幅度。

10.8 解释联合形状和纹理编码算法。在因特网上通常用什么方法？

10.9 如何扩展像素方式的形状编码器，使它不仅能在图像编码器中也能在视频编码器中有效地工作？

10.10 任意形状物体的纹理编码方法有哪些？

10.11 两种主要的指导基于区域编码器设计的假设是什么？

10.12 基于物体的分析合成编码与基于区域的编码有什么区别？哪种编码器对自然视频序列更适合？哪种对卡通更适合？解释原因。

10.13 对OBASC最常用的信源模型是什么？

10.14 模型失效检测忽略几何失真。几何失真指的是什么？它们会随时间积累吗？

10.15 解释用滤波器级联的图像合成和用位移矢量场级联的图像合成。哪一种方法的效率更高？

10.16 为什么基于知识的编码器比OBASC编码器能更有效地编码场景？至少指出两种

原因。

10.17　为什么语义编码是非常有效的？为什么我们目前还没有语义编码器？

10.18　动作单元和脸部动画参数之间的区别是什么？

10.19　解释分层编码器的各层。目前通常使用哪些层？

10.11　文献目录

[1] Aach, T., and A. Kaup. Bayesian algorithms for change detection in image sequences using Markov random fields. *Signal Processing: Image Communication* (Aug. 1995), 7(2):147–60.

[2] Aizawa, K., H. Harashima, and T. Saito. Model-based analysis-synthesis image coding (MBASIC) system for a person's face. *Signal Processing: Image Commun.* (Oct. 1989), 1(2):139–52.

[3] Bartlett, M. S., et al. Classifying facial action. In D. Touretzky, M. Mozer, and M. Hasselmo, eds., *Advances in Neural Information Processing Systems 8*, 823–29. Cambridge, MA: MIT Press, 1996.

[4] Born, G. *The File Formats Handbook*. Boston: International Thompson Computer Press, 1995.

[5] Brady, N., and F. Bossen. Shape compression of moving objects using context-based arithmetic encoding. *Signal Processing: Image Communications* (May 2000), 15(7–8):601–17.

[6] Chen, T., C. T. Swain, and B. G. Haskell. Coding of subregions for content-based scalable video. *IEEE Trans. on Circuits and Systems for Video Technology* (Feb. 1997), 7:256–60.

[7] Choi, C. S., K. Aizawa, H. Harashima, and T. Takebe. Analysis and synthesis of facial image sequences in model-based image coding. *IEEE Trans. on Circuits and Systems for Video Technology* (June 1994), 4(3):257–75.

[8] Chowdury, M. F., A. F. Clark, A. C. Downton, E. Morimatsu, and D. E. Pearson. A switched model-based coder for video signals. *IEEE Trans. on Circuits and Systems for Video Technology* (June 1994), 4(3):216–27.

[9] de Klerk, M., R. Prasad, J. H. Bons, and N. B. J. Weyland. Introducing high-resolution line graphics, in UK teletext using differential chain coding. *Proceedings of IEE, Part I* (1990), 137(6):325–34.

[10] Eckman, P., and V. W. Friesen. *Facial Coding System*. Palo Alto, CA: Consulting Psychologists Press, 1977.

[11] Eden, M., and M. Kocher. On the performance of a contour coding algorithm in the context of image coding. Part I: Contour segment coding. *Signal Processing* (1985), 8:381–86.

[12] Eisert, P., T. Wiegand, and B. Girod. Model-aided coding: A new approach to incorporate facial animation into motion-compensated video coding. *IEEE Trans. on Circuits and Systems for Video Technology* (April 2000), 10(3):344–58.

[13] Forchheimer, R., O. Fahlander, and T. Kronander. A semantic approach to the transmission of face images. *International Picture Coding Symposium (PCS'84)* (July 1984), p. 10.5.

[14] Freeman, H. On the encoding of arbitrary geometric configurations. *IRE Trans. Electronic Computers* (June 1961), EC-10(2):260–68.

[15] Gerken, P. Object-based analysis-synthesis coding of image sequences at very low bit rates. *IEEE Trans. Circuits and Systems for Video Technology* (June 1994), 4:228–35.

[16] Gilge, M., T. Engelhardt, and R. Mehlan. Coding of arbitrarily shaped image segments based on a generalized orthogonal transform. *Signal Processing: Image Communications* (Oct. 1989), 1(2):153–80.

[17] Graphics interchange format, version 89a. Technical report, CompuServe Incorporated, Columbus, Ohio, July 1990.

[18] Hötter, M. Object-oriented analysis-synthesis coding based on two dimensional objects. *Signal Processing: Image Communication* (Dec. 1990), 2(4):409–28.

[19] Hötter, M. Optimization and efficiency of an object-oriented analysis-synthesis coder. *IEEE Trans. on Circuits and Systems for Video Technology* (April 1994), 4(2):181–94.

[20] ISO/IEC. IS 14496-1: Information technology—coding of audio-visual objects—part 1: Systems, 1999. (MPEG-4 Systems).

[21] ISO/IEC. IS 14496-2: Information technology—coding of audio-visual objects—part 2: Visual, 1999. (MPEG-4 Video).

[22] ITU-T. Recommendation H.263: Video coding for low bit rate communication, 1998.

[23] ITU-T (International Telecommunication Union, Telecommunication Standardization Sector). Recommendation T.6: Facsimile coding schemes and coding control functions for group 4 facsimile apparatus, Geneva, 1988.

[24] ITU-T. Recommendation T.82 (03/93)—information technology—coded representation of picture and audio information—progressive bi-level image compression, Geneva, 1993.

[25] Jain, A. K. *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.

[26] Jordan, C. L., F. Bossen, and T. Ebrahimi. Scalable shape representation for content based visual data compression. In *IEEE Int. Conf. Image Process. (ICIP'97), Special Session on Shape Coding*, Santa Barbara, CA, 1997, 1:512–15.

[27] Kalra, P., A. Mangili, N. Magnenat-Thalmann, and D. Thalmann. Simulation of facial muscle actions based on rational free form deformations. *Eurographics* (1992), 59–69.

[28] Kampmann, M., and J. Ostermann. Automatic adaptation of a face model in a layered coder with an object-based analysis-synthesis layer and a knowledge-based layer. *Signal Processing: Image Communications* (March 1997), 9(3): 201–20.

[29] Kaneko, T., and M. Okudaira. Encoding of arbitrary curves based on the chain code representation. *Commun.* (1985), 33:697–707.

[30] Katsaggelos, A. K., L. P. Kondi, F. W. Meier, J. Ostermann, and G. M. Schuster. MPEG-4 and rate distortion based shape coding techniques. *Proceedings of the IEEE, Special Issue on Multimedia Signal Processing, Part Two* (June 1998), 86(6):1126–54.

[31] Kaup, A. A new approach towards description of arbitrarily shaped image segments. *SPIE Conf. Visual Communications and Image Processing (VCIP'97)* (Feb. 1997), SPIE-3012:731–41.

[32] Kim, J. I., A. C. Bovik, and B. L. Evans. Generalized predictive binary shape coding using polygon approximation. *Signal Processing: Image Communications*

(May 2000), 15(7–8):643–63.

[33] Koplowitz, J. On the performance of chain codes for quantization of line drawings. *IEEE Trans. Pattern Anal. Machine Intell.* (March 1981), 3(2):180–85.

[34] Kunt, M., A. Ikonomopoulos, and M. Kocher. Second-generation image-coding techniques. *Proceedings of the IEEE* (April 1985), 73:549–74.

[35] Li, H., P. Roivainen, and R. Forchheimer. 3-D motion estimation in model-based facial image coding. *IEEE Trans. Pattern Anal. Machine Intell.* (June 1993), 15(6):545–555.

[36] Li, S., and W. Li. Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding. *IEEE Trans. on Circuits, Systems, and Video Technology* (Aug. 2000), 10(5):725–43.

[37] Martinez, G. Shape estimation of articulated 3-D objects for object-based analysis-synthesis coding (OBASC). *Signal Processing: Image Communications* (March 1997); 9(3):175–99.

[38] Musmann, H. A layered coding system for very low bit rate video coding. *Signal Processing: Image Communications* (Nov. 1995), 7(4-6):267–78.

[39] Musmann, H. G., M. Hötter, and J. Ostermann. Object oriented analysis-synthesis coding of moving images. *Signal Processing: Image Communications* (Oct. 1989), 1:117–38.

[40] Netravali, A. N., and B. G. Haskell. *Digital Pictures—Representation and Compression.* New York: Plenum Press, 1988.

[41] Neuhoff, D. L., and K. G. Castor. A rate and distortion analysis of chain codes for line drawings. *IEEE Transactions on Information Theory* (1985), 31(1): 53–67.

[42] Nunes, P., F. Pereira, and F. Marques. Multi-grid chain coding of binary shapes. *IEEE Int. Conf. Image (ICIP'97),* Santa Barbara, CA, 1997, III:114–17.

[43] O'Connell, K. J. Object-adaptive vertex-based shape coding method. *IEEE Transactions on Circuits and Systems* (Feb. 1997), 7(1):251–55.

[44] Ostermann, J. The block-based coder mode in an object-based analysis-synthesis coder. *28th Asilomar Conference on Signals, Systems, and Computers* (Nov. 1994), TP 6–1.

[45] Ostermann, J. Object-based analysis-synthesis coding (OBASC) based on the source model of moving flexible 3-D objects. *IEEE Trans. on Image Processing,* (September 1994), 3(5):705–11.

[46] Ostermann, J. Object-based analysis-synthesis coding (OBASC) based on the source model of moving rigid 3-D objects. *Signal Processing: Image Communications* (1994), 6(2):143–61.

[47] Ostermann, J. Segmentation of image areas changed due to object motion considering shadows. In Y. Wang, et al., eds., *Multimedia Communications and Video Coding,* 241–246. New York: Plenum Press, 1996.

[48] Pratt, W. K. *Digital Image Processing.* 2nd ed. New York: John Wiley & Sons, 1991.

[49] Rydfalk, R. Candide, a parameterised face. Internal Report Lith–ISY–I–0866, Linköping University, Linköping, Sweden, 1987.

[50] Saghri, J. A., and H. Freeman. Analysis of the precision of generalized chain codes for the representation of planar curves. *IEEE Trans. Pattern Anal. Machine Intell.* (Sept. 1981), 3(5):533–39.

[51] Salembier, P., and F. Marques. Region-based representations of image and video: segmentation tools for multimedia services. *IEEE Trans. on Circuits and Systems for Video Technology* (Dec. 1999), 9(8):1147–69.

[52] Salembier, P., F. Marques, and A. Gasull. Coding of partition sequences. In L. Torres and M. Kunt, eds., *Video Coding: The Second Generation Approach*, 125–70. Boston: Kluwer Academic Publishers, 1996.

[53] Salembier, P., L. Torres, F. Meyer, and C. Gu. Region-based video coding using mathematical morphology. *Proceedings of the IEEE, Special Issue on Digital TV* (June 1995), 83(6):843–57.

[54] Schuster, G. M., and A. G. Katsaggelos. An optimal segmentation encoding scheme in the rate-distortion sense. *IEEE Int. Symp. Circuits and Systems (ISCAS'96)*, Atlanta, 1996, 2:640–43.

[55] Sikora, T., S. Bauer, and B. Makai. Efficiency of shape-adaptive transforms for coding of arbitrarily shaped image segments. *IEEE Trans. on Circuits and Systems for Video Technology* (June 1995), 5(3):254–58.

[56] Stauder, J., R. Mech, and J. Ostermann. Detection of moving cast shadows for object segmentation. *IEEE Transactions on Multimedia* (March 1999), 1(1):65–76.

[57] Tekalp, A. M., and J. Ostermann. Face and 2-D mesh animation in MPEG-4. *Signal Processing: Image Communications, Special Issue on MPEG-4* (Jan. 2000), 15:387–421.

[58] Terzopolous, D., and K. Waters. Physically-based facial modeling, analysis and animation. *Journal of Visualization and Computer Animation* (1990), 1:73–80.

[59] Torres, L., and M. Kunt. Second Generation Video Coding Techniques. In L. Torres and M. Kunt, eds., *Video Coding: The Second Generation Approach*, 1–30. Boston, MA: Kluwer Academic Publishers, 1996.

[60] van Otterloo, P. *A Contour-Oriented Approach for Shape Analysis*. Hemel Hampstead, England: Prentice Hall International, 1991.

[61] Zahn, C. T., and R. Z. Roskies. Fourier descriptors for plane closed curves. *IEEE Transactions on Computer* (March 1972), 21(3):269–81.

第11章 可分级视频编码

在前面两章中介绍的编码方法包括基于波形和基于内容的方法，目的是对于固定的比特率使编码效率最优化。当许多用户试图通过不同的通信链路接入相同的视频时，这会出现困难。例如，可以在通过高速链路(例如ADSL调制解调器)连接到服务器的终端上实时下载以1.5 Mbps编码的MPEG-1视频并重放。但仅有56 kbps调制解调器连接的用户将不能实时接收足够的比特进行重放。可分级性是指通过仅解码一部分压缩的比特流物理地恢复有意义的图像或视频信息的能力。在前面一个例子中，如果视频流是可分级的，那么具有高带宽连接的用户可以下载整个比特流以观看全质量的视频，而具有56 kbps连接的用户将只下载流的一个子集，观看一个低质量的演播。

这个例子只说明了可分级性的一方面：带宽可分级性。可分级流也可以提供对变化的信道误差特征的适应性，以及对接收终端计算能力的适应性。对于无线通信，可分级性允许调整信源码率，并允许应用非平衡误码保护以适应信道误码条件。对于因特网传输，可分级性使可变比特率传输，有选择地丢弃比特，以及相应于不同的调制解调器速率、变化的信道带宽和不同的设备能力调整信源码率成为可能。当我们进入无线、因特网和多媒体的汇合处时，可分级性对于从任何地方、由任何人、在任何时间、用任何设备和以任何形式进行丰富的多媒体访问就变得愈发重要了。

可分级编码器可有粗间隔度(在两层或三层内——这些也称为分层编码器)或细间隔度。在细间隔度的极端情况下，比特流可以在任何点被截断。保留的比特越多，重建图像的质量就越好。我们称这样的比特流为嵌入式的。嵌入式编码器能进行精确的比特率控制，这在许多应用系统中都是所希望的特性。例如，网络滤波器(在第15章中描述)可以从嵌入式比特流中选择传输的比特数以匹配可用带宽。

可分级编码一般是通过提供一个视频的多种版本实现的，这些版本是就幅度分辨率(称为质量可分级性或SNR可分级性)、空间分辨率(空间可分级性)、时间分辨率(时间可分级性)、频率分辨率(频率可分级性，更经常称为数据分割)或这些选项的组合而言的。图11.1示出一组从可分级比特流中解码的图像；它是通过联合质量可分级性与空间可分级性得到的。

可分级内容可以在帧级或物体级进行访问。后者是指基于物体的可分级性，如在MPEG-4标准中所定义的[5]。在本节中，我们首先介绍实现可分级性的4个基本方案，包括质量、空间、时间和频率可分级性(11.1节)。然后描述如何在物体级实现可分级性(11.2节)。尽管类似的概念可以应用于不同类型的编码器，但我们将把讨论集中于修正的基于块的混合编码器(9.3.1节)，以实现各种可分级性模式。最后，我们描述基于小波的编码方法，由小波变换的性质，它自然导致细间隔度可分级性。

注意，应付变化的信道环境和接收机能力的另一种方法是通过同时联播，它简单地把同一视频编码几次，每次具有不同的质量或分辨率设置。这种方法尽管简单，但效率很低，因为一

个较高质量或分辨率的比特流实际上重复了已经包含在较低质量或分辨率比特流中的信息，以及一些附加信息。另一方面，为了提供可分级功能，与目前的不可分级编码器相比较，编码器必须牺牲一定的编码效率。可分级编码的设计目标是在实现可分级性要求的同时使编码效率的降低达到最小。

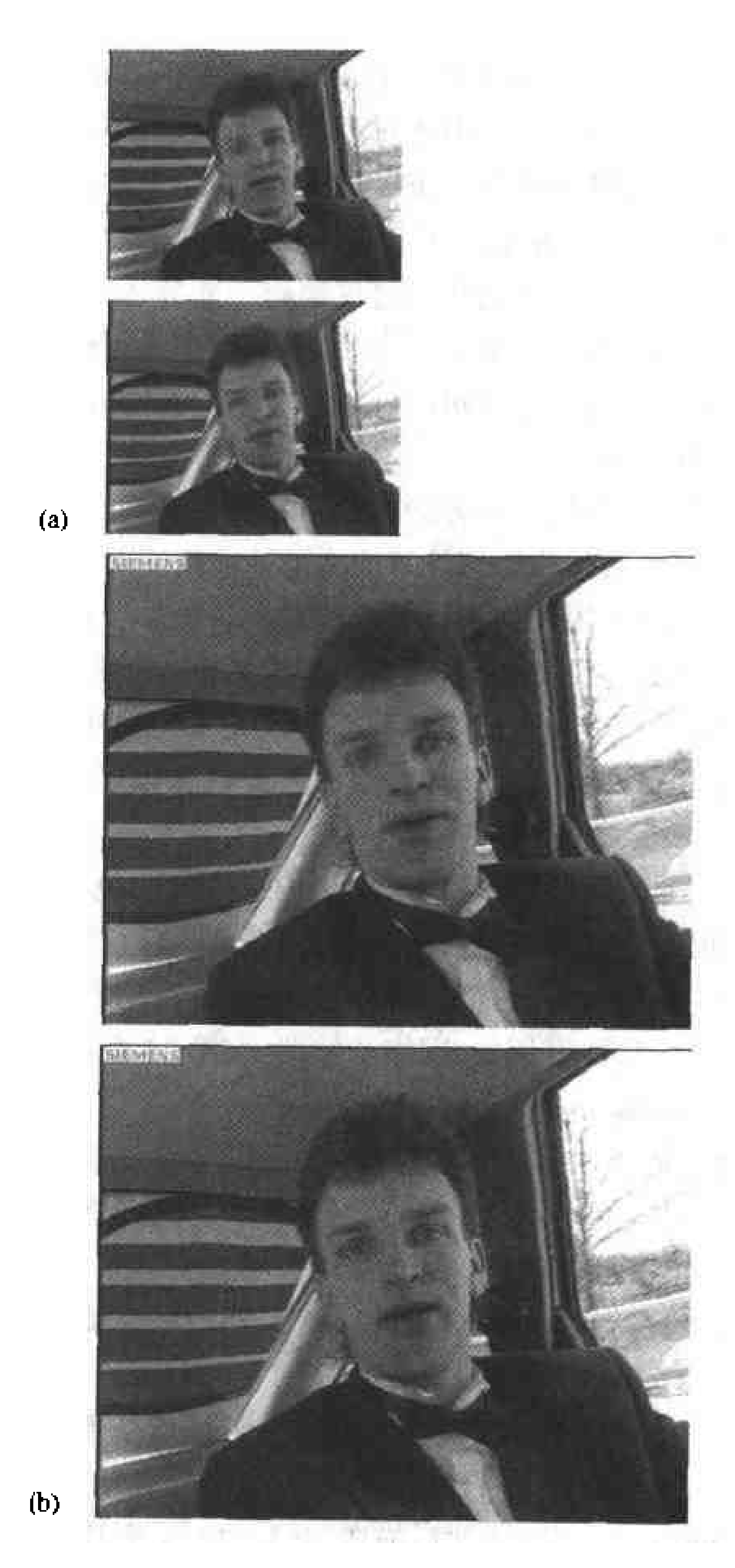


图 11.1 从联合了质量和空间可分级性的可分级比特流中解码的视频帧：(a)中两幅图像的尺寸为 176×144，分别是在 6.5 kbps 和 133.9 kbps 下解码得到的；(b)中两幅图像的尺寸为 352×288，分别是在 21.6 kbps 和 436.3 kbps 下解码得到的

11.1 可分级性的基本模式

11.1.1 质量可分级性

质量可分级性定义为具有可变的彩色模式精度的视频序列的表示。这一般是通过以越来越精细的量化步长量化彩色值(在原始或变换域中)实现的。因为不同的量化精度导致原始的与量化的视频之间不同的 PSNR,所以这类可分级性通常称为 SNR 可分级性。

图 11.2 示出了一个具有 N 层质量可分级性的比特流。解码第一层(也称为基本层)提供一个低质量的重建图像版本。进一步解码其余的层(也称为增强层)导致重建图像的质量提高,直到最高质量。第一层是通过对原始图像或在变换(例如 DCT)域中应用一个粗糙的量化器得到的。第二层包含原始图像与由第一层重建的图像之间的量化差值,使用的量化器比用于产生第一层的量化器更精细。类似地,后面的每一层包含原始图像与由前面一层重建的图像之间的量化差值,量化中使用愈加精细的量化器。

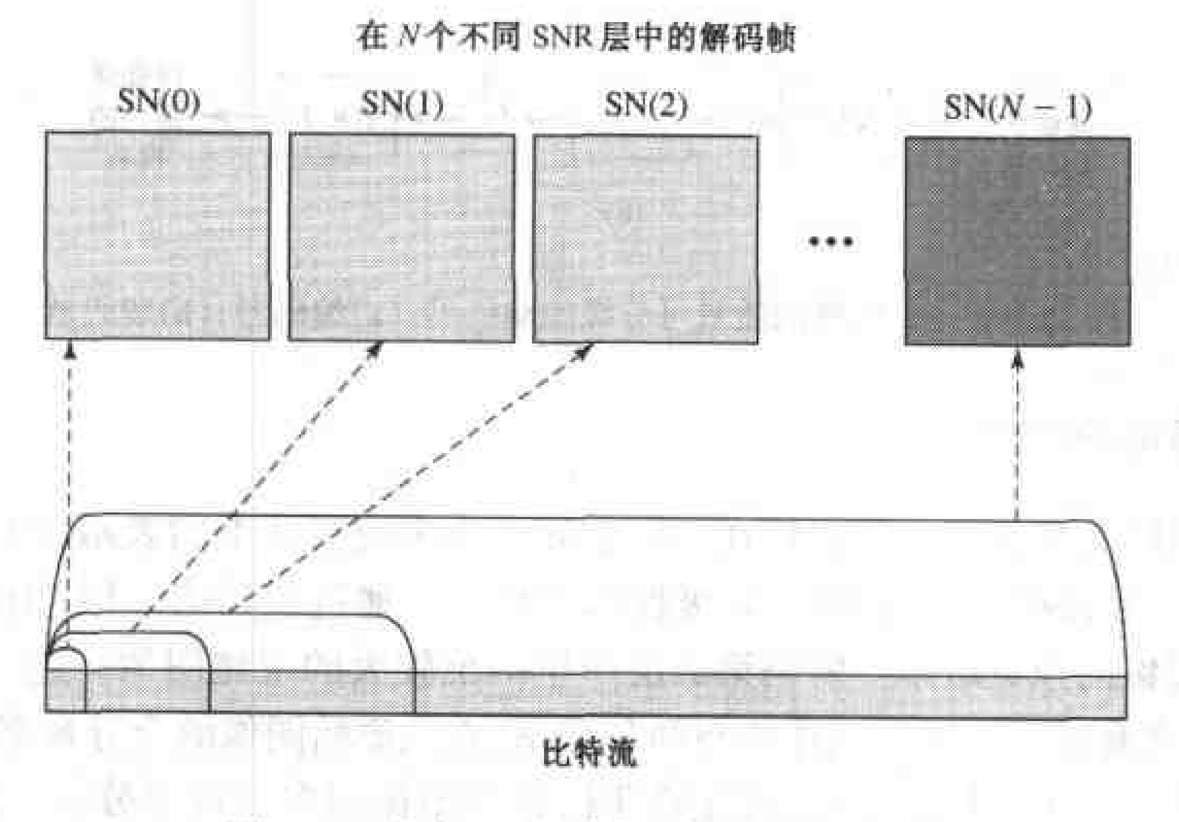


图 11.2　具有 N 层质量可分级性的比特流

图 11.3(a)画出了一个具有两级质量可分级性的编码器。对于基本级,编码器以与典型的基于块的混合编码器一样的方式工作。对于增强级,按如下方式工作:

1. 在基本级对原视频帧(或运动补偿误差帧)进行 DCT 变换并量化;
2. 用反量化重建基本级 DCT 系数;
3. 从原 DCT 系数中减去基本级的 DCT 系数;
4. 用小于基本级的量化参数量化该残差;
5. 用 VLC 编码量化比特。

由于增强级使用了较小的量化参数,它能够达到比基本级更好的质量。

图 11.3(b)画出了解码器的工作过程。对于基本级,解码器与不可分级视频解码器的工作完全一样。对于增强级,必须接收到两级,用可变长解码(VLD)进行解码,并进行反量化。

然后把基本级的 DCT 系数值加到增强级的精细的 DCT 系数上。这一步之后，对求和后的 DCT 系数进行 DCT 逆变换，产生增强级的解码视频。

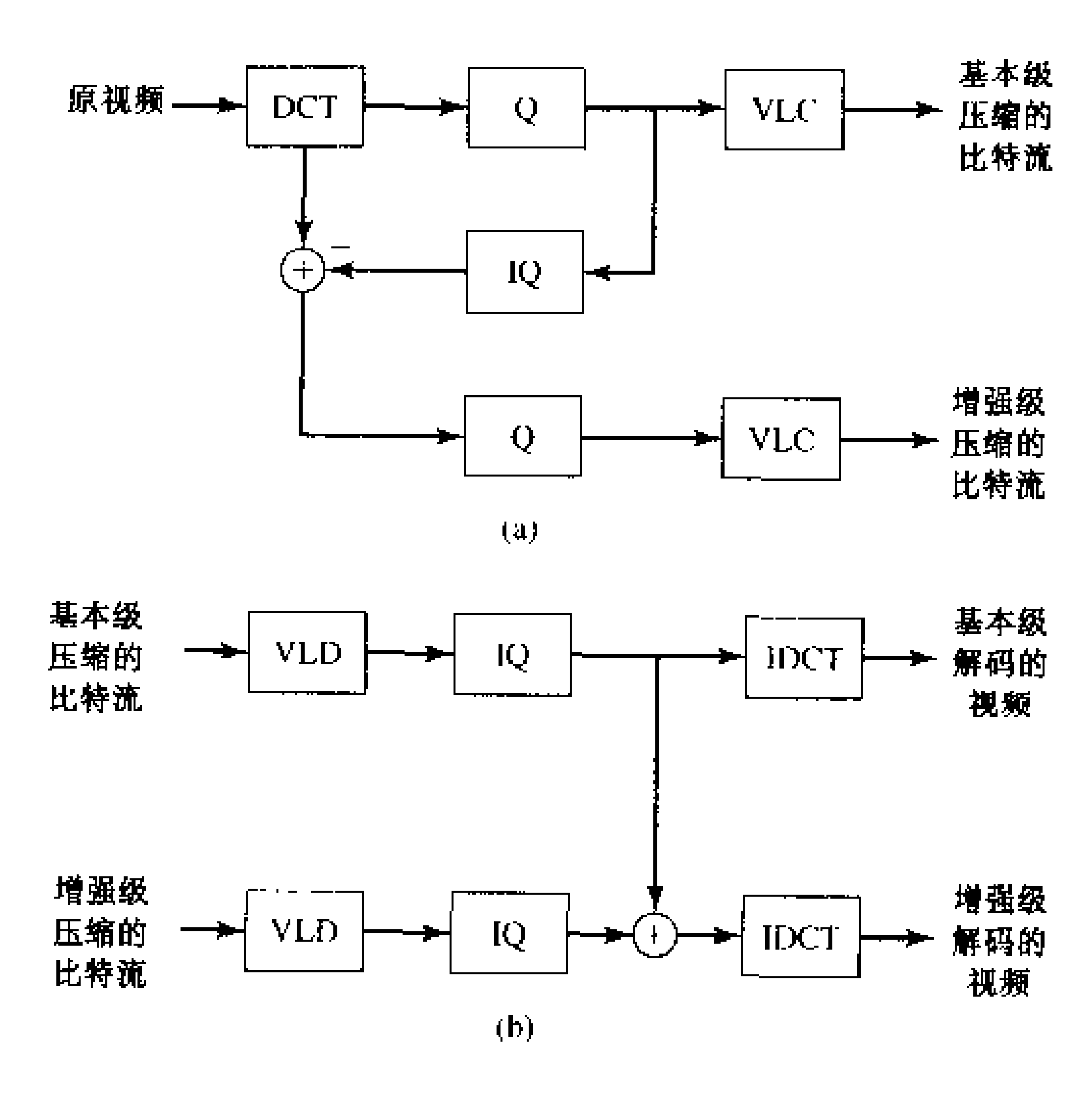


图 11.3 一个两级的质量可分级编解码器：(a)编码器；(b)解码器

11.1.2 空间可分级性

空间可分级性定义为同一个视频在不同空间分辨率或尺寸下的表示(图 11.4(a)和(b))。图 11.5 说明了一个具有 M 层空间可分级性的比特流。通过解码第一层，用户可以显示一个低分辨率解码图像的预览版本。解码第二层产生一个较大的重建图像。进一步地，通过逐级解码其余的层，观看者可以增加图像的空间分辨率，直到原始图像的全分辨率。

为了产生这样一个分层比特流，首先要获得原始图像的多分辨率分解。直接编码最低分辨率的图像以产生第一层(即基本层)。为了产生第二层，先把由第一层解码的图像内插到第二个最低的分辨率，并且在这个分辨率下对原始图像与内插图像之间的差进行编码。以相同方式产生每一个后续分辨率的比特流：首先在该分辨率下基于前面的层形成一个估计图像，然后在该分辨率下编码估计图像与原始图像之间的差。

图 11.6(a)示出了一个两层的空间可分级编码器的方框图。对于基本层，原视频首先进行空间下采样[①]，然后进行 DCT 变换、量化和 VLC 编码。对于增强层，进行以下的工作：

1. 在基本层，对原视频进行空间下采样、DCT 变换和量化；
2. 通过反量化和逆 DCT 重建基本层图像；

① 例如，通过将每 2×2 个像素用它们的平均值替换，可以实现 4:1 比率的空间下采样。用更复杂的预滤波器可以减少下采样图像的重叠效应，但以增加复杂度作为代价。

3. 对基本层图像进行空间上采样[1]；
4. 从原始图像中减去上采样的基本层图像；
5. 对残差进行 DCT 变换，并用小于基本层的量化参数进行量化；
6. 用 VLC 编码量化的比特。

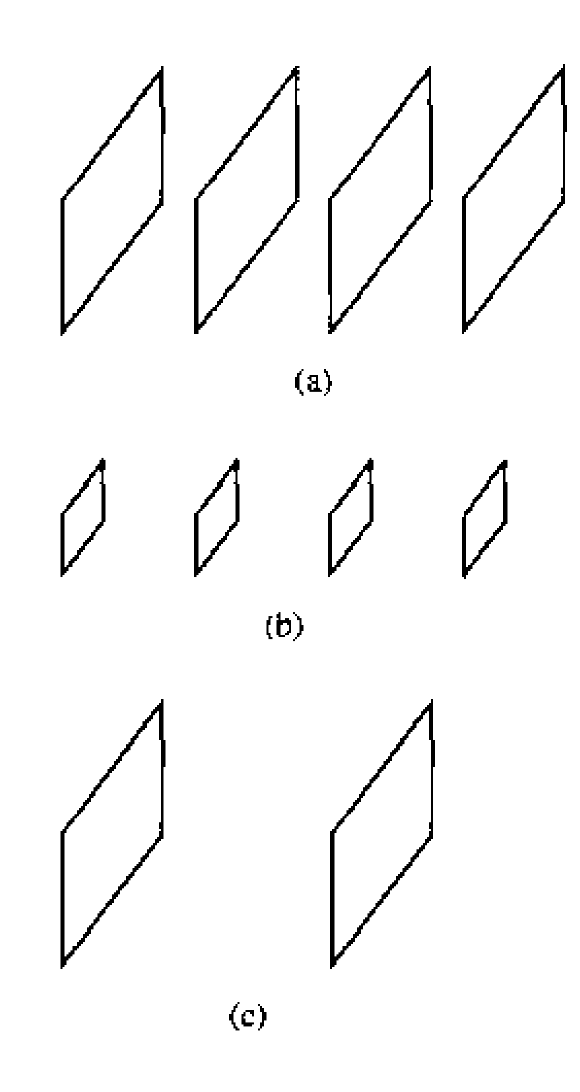


图 11.4　视频流的空间和时间分级：(a)原始视频帧；(b)帧缩小到原始尺寸的1/4；(c)时间上分级的帧

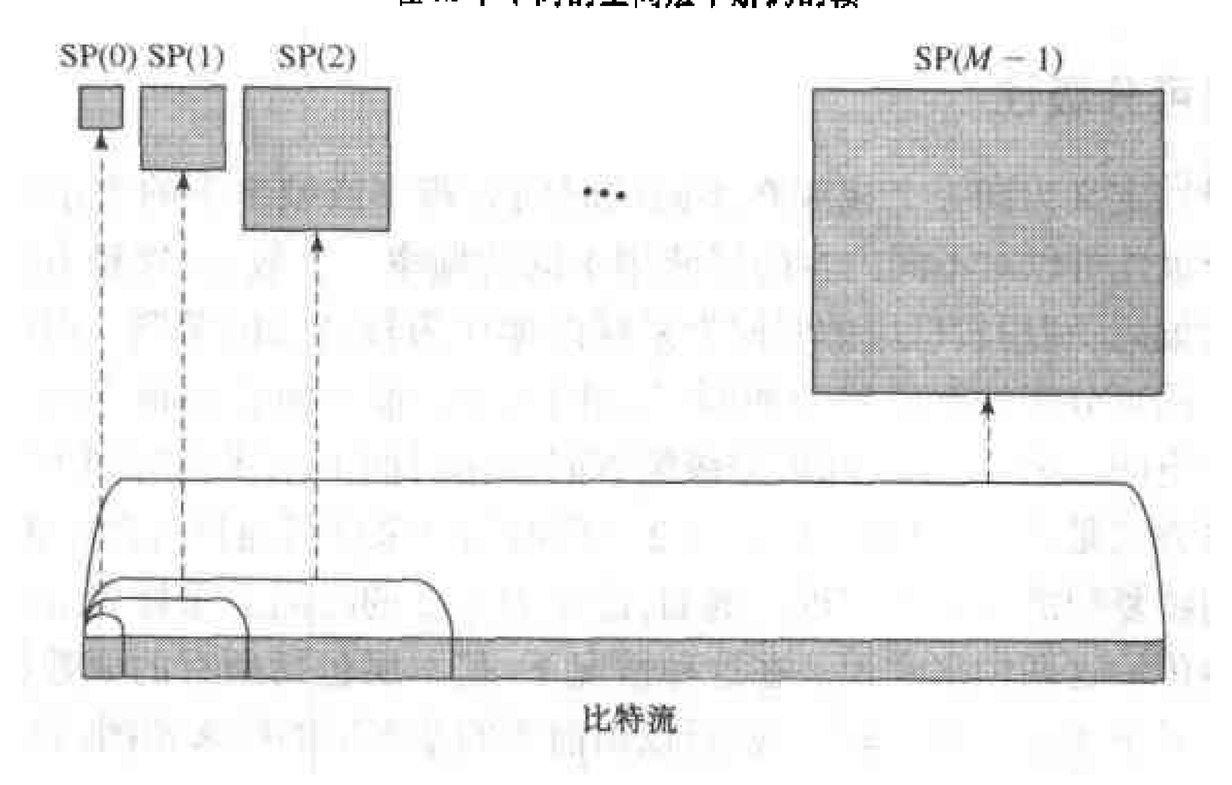


图 11.5　具有 M 层空间可分级性的比特流

[1] 例如，一个实现 1:4 比率的空间上采样的简单方法是每个像素复制四次。

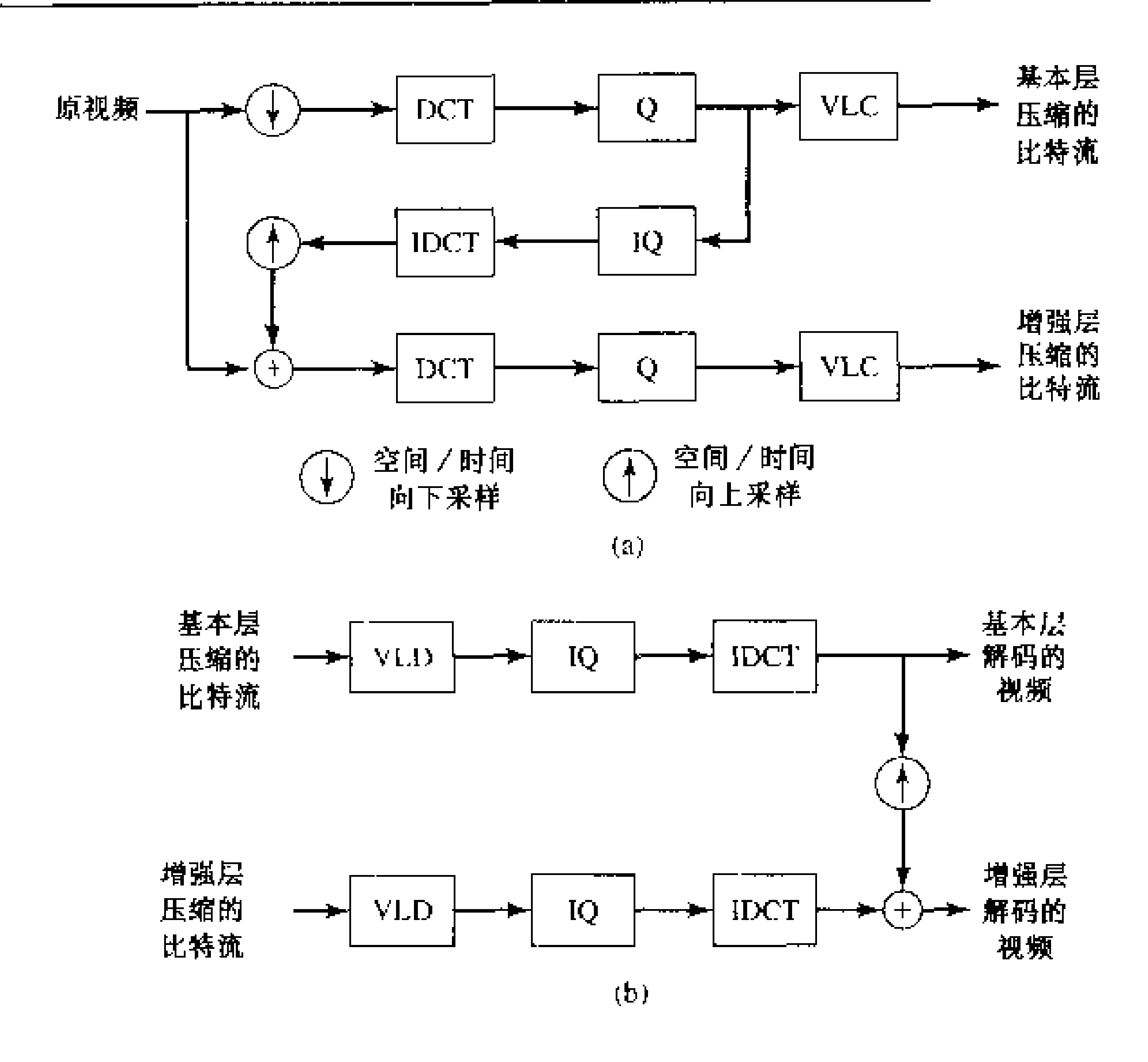


图 11.6 一个两层的空间/时间可分级编解码器:(a)编码器;(b)解码器

由于增强层使用了较小的量化参数,它可以达到比基本层更高的质量。

图 11.6(b)画出了一个具有两层可分级性的空间可分级解码器。对于基本层,解码器的工作与不可分级的视频解码器完全一样。对于增强层,必须接收到两层,用 VLD 解码,进行反量化和逆 DCT 变换。然后上采样基本层图像。把上采样的基本层图像与增强层的细节相结合形成增强层解码视频。

11.1.3 时间可分级性

时间可分级性定义为同一个视频在不同的时间分辨率或帧率下的表示(见图 11.4(a)和(c))。时间可分级性可以对不同内容的层使用不同的帧率。一般,以这种方法对时间可分级视频进行有效的编码:利用较低层的时间上采样图像作为较高层的预测。时间可分级编解码器的方框图与空间可分级编解码器的相同(见图 11.6)。惟一的差别是空间可分级编解码器用空间下采样和空间上采样,而时间可分级编解码器用时间下采样和时间上采样。进行时间下采样的最简单方法是跳帧。例如,比率为 2:1 的时间下采样可通过每两帧丢弃一帧来实现。时间上采样可用帧复制的方法来实现。例如,比率为 1:2 的时间上采样可通过每帧复制一个副本并在下一步传输这两帧来实现。在这种情况下,基本层包括所有的偶数帧而增强层包括所有的奇数帧。对于运动补偿,基本层帧将仅由前面的基本层的帧来预测,而增强层的帧由基本层的帧和增强层的帧都可以预测。

11.1.4 频率可分级性

用多个层表示视频帧的另一种方法是通过在每一层包含不同的频率分量,基本层包含低

频分量而其他层包含逐渐增加的较高频率分量。在这种方法中,基本层将提供一个模糊的图像,加上各增强层将产生逐渐清晰的图像。可通过整帧变换,例如子带分解或小波变换,或通过基于块的变换(例如块 DCT)来实现这种分解。在基于块的混合编码器中实现这个思想的一种方法是,在基本层中包括模式信息、运动信息和每个宏块的前几个 DCT 系数,而在增强层中包括其余的 DCT 系数。在 MPEG-2 标准中,这称为数据分割。我们将在 11.3 节中讨论基于小波的可分级编码器。

11.1.5 基本方案的组合

质量、空间、时间和频率可分级性是基本的可分级机制。可把它们组合起来以达到较细的间隔度。图 11.7 示出了一个空间和质量可分级性组合的情况。在这个例子中,比特流由 M 个空间层组成,每个空间层包括 N 级质量可分级性。在这种情况下,渐进地传输并解码比特流可以提高重建图像的空间分辨率和质量。次序是在一个给定的空间分辨率下提高图像质量,直到达到该分辨率下的最佳质量,然后把空间分辨率提高到较高的级并再次提高质量。

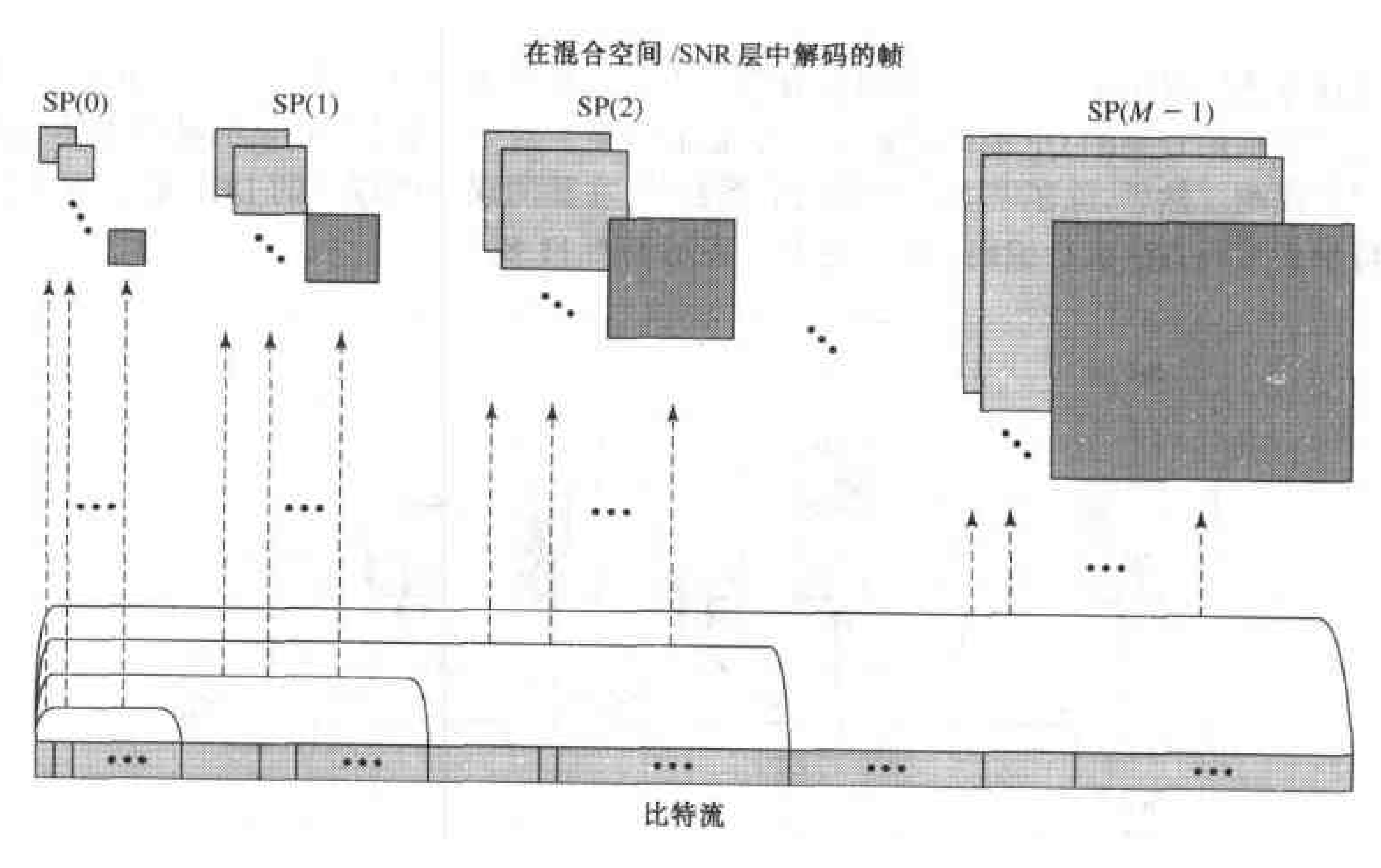


图 11.7　$N \times M$ 层组合的空间/时间可分级性

11.1.6 细间隔度可分级性

我们已描述的可分级性方法生成由若干层组成的比特流,基本层后面是一个或几个增强层。这种类型的编码器仅提供粗间隔度,这只在码率以大的离散步长增加时才能获得质量的改善。如果在一个特殊的层中仅接收到部分比特,则再生的视频质量很大程度上取决于在前面层中接收到的比特,而不会受益于从该层接收到的比特。细间隔度可分级性(FGS, Fine-granularity scalability)是指一种编码方法,通过这种方法,码率和质量以小得多的步长增加。在极限情况下,比特流可以通过每一个附加的比特提供连续的视频质量的改善,这种编码方法称为嵌入式编码。很明显,FGS 和嵌入式编码比其他可分级方法能更有效地适应实际网络中的带宽变化。

实际上,比特流完全嵌入或具有细间隔度的要求经常是宽松的。可以首先生成基本层,以提供一个低的但有保证的质量水平,然后可生成增强层以提供细间隔度。例如,在 MPEG-4 的 FGS 模式[14,11]中采用了这种方法。在这种情况下,采用传统的基于块的混合编码方法,以一个相对大的 QP,生成一个给定帧率下的基本层流。然后,对于每个编码帧(帧内或帧间编码的),把原始 DCT 系数[①] 与基本层中的量化系数之间的差(称为细节系数)编码成细间隔度流。这是通过用非常小的 QP 来量化细节系数[②],再通过连续的比特平面编码表示量化索引实现的。

具体地说,每个块中量化的细节系数的绝对值用二进制表示来确定。从包含非零比特的最高比特平面开始,用游程编码相继地编码每个比特平面,逐块地进行。可以用哈夫曼或算术编码对游程编码。一般,不同的比特平面采用不同的码书,因为在不同的平面上游程分布是不同的。当只有增强层流的部分集被解码时,取决于保留的比特中包括多少比特平面,重建视频的质量将处于仅由基本层获得的质量到通过细节 DCT 系数所用的 QP 获得的质量之间。流的间隔度是在比特平面级:每一个增加的完整的比特平面将引起 DCT 系数精度改善 2 倍[③]。

这种方法仅提供在固定的空间时间分辨率下的质量可分级性。为了增加时间可分级性,基本层可包括用较高的 QP 编码的帧的一个亚采样集。对于增强层,其余的帧由基本层的帧进行双向预测。然后,除了基本层帧的细节系数外,在预测误差图像中的 DCT 系数被量化,并用比特平面编码方法进行编码[22]。这个方案示于图 11.8。

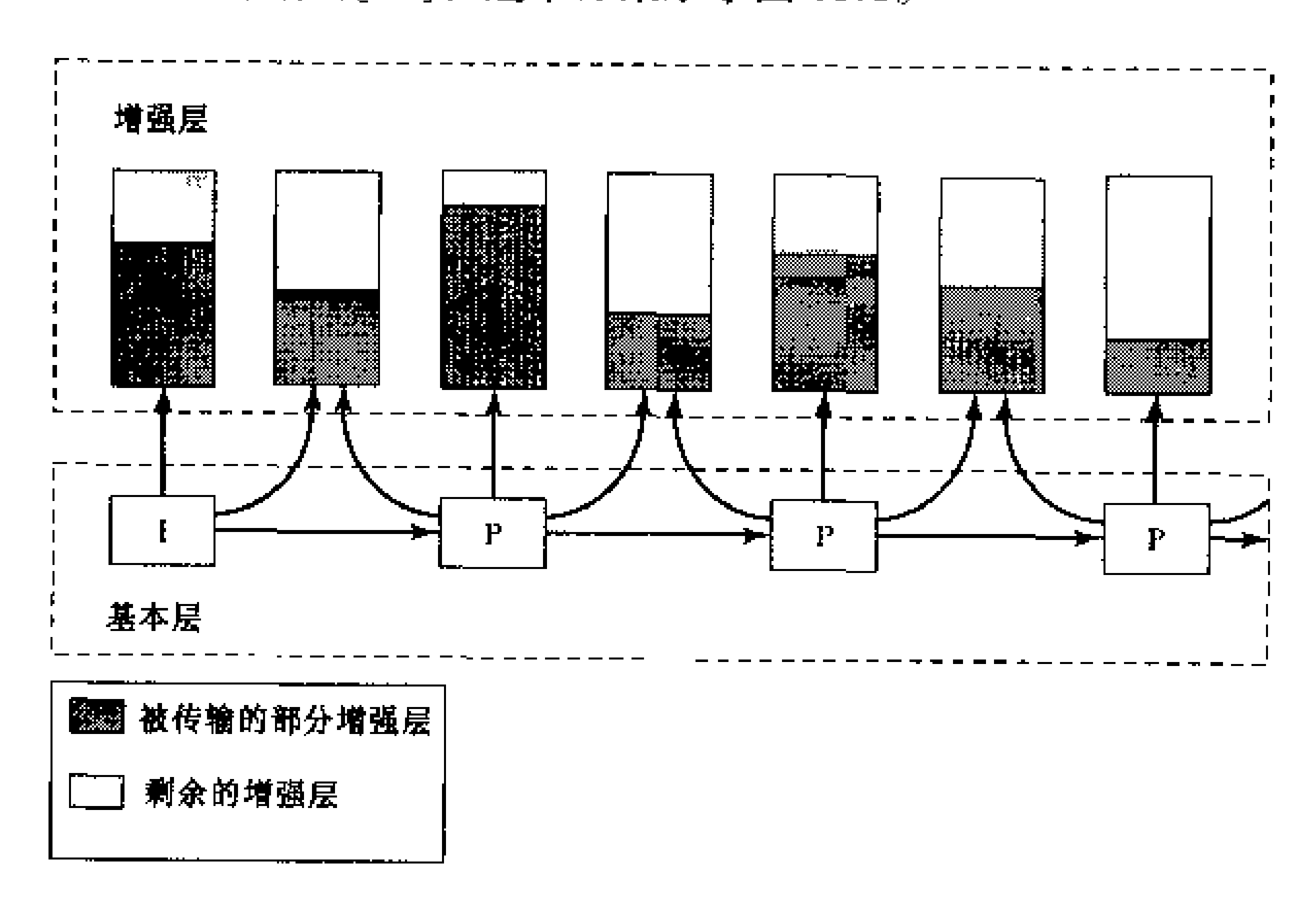


图 11.8 支持联合质量时间可分级性的增强层流的部分传输图示

这个 FGS 方案的局限性是必须完整地传送基本层,并且是没有差错的。在实际网络中这

① 对于每个块,变换是加于原始的块或运动补偿误差块上,这取决于该块在基本层中是以帧内模式还是以帧间模式编码的。

② 如果希望得到在最高码率下的无损表示,就不进行量化。

③ 事实上,对于游程表示可达到码字级的间隔度,因为每个所收到的完整的附加码字可提高某些系数的重建精度(尽管取决于可用带宽),更实际的做法是在比特平面级截取 FGS 流。

有可能做到也可能做不到。另一个问题是，当基本层比特率保持很低时(以便增加比特流的可分级范围)，FGS 方法的编码效率与不可分级编码器比较将被显著地降低。这是因为重建的基本层的帧是作为运动补偿预测的参考帧使用的。尽管这种选择消除了增强层中的时间误码扩散问题，但在基本层的帧中的高量化误差减少了预测增益。提高 FGS 方法编码效率的一种方法是周期地使用中间的增强层(由某些但不是所有的比特平面重建的)作为运动补偿预测的参考[24]。

用基于块的变换完全实现嵌入式编码是困难的。我们将在 11.3 节中说明如何用小波变换达到这个目的。

11.2　基于物体的可分级性

在前一节中介绍的各种可分级编码方法在整个视频帧上进行相同的工作。在基于物体的时间可分级性(OTS)中，所选择物体的帧率被提高，以便它具有比其余的区域更平滑的运动。(换句话说，所选择物体的帧率比其余区域的帧率高。)下面我们将主要介绍基于 MPEG-4 实现的 OTS。

在 OTS 中有两类增强结构。图 11.9 示出了一个类型 1 的例子，其中 VOL0(视频物体层 0)是具有一个物体和一个背景的整个帧，而 VOL1 表示 VOL0 中的特殊物体。以低帧率编码 VOL0，而以高于 VOL0 的帧率编码 VOL1。在这个例子中，帧 2 和 4 是由 VOL0 的帧 0 预测编码的，然后把增强层的物体重叠到这个组合的帧。图 11.10 示出了类型 1 的另一个例子，它采用双向预测形成增强层中的双向 VOP(B-VOP，双向视频物体平面)。在这种情况下，VOL1 中的帧 2 和 4 是由 VOL0 中的帧 0 和 6 预测的。在这两种情况下，两个附加的形状数据(前向形状和后向形状)被编码以执行背景合成。

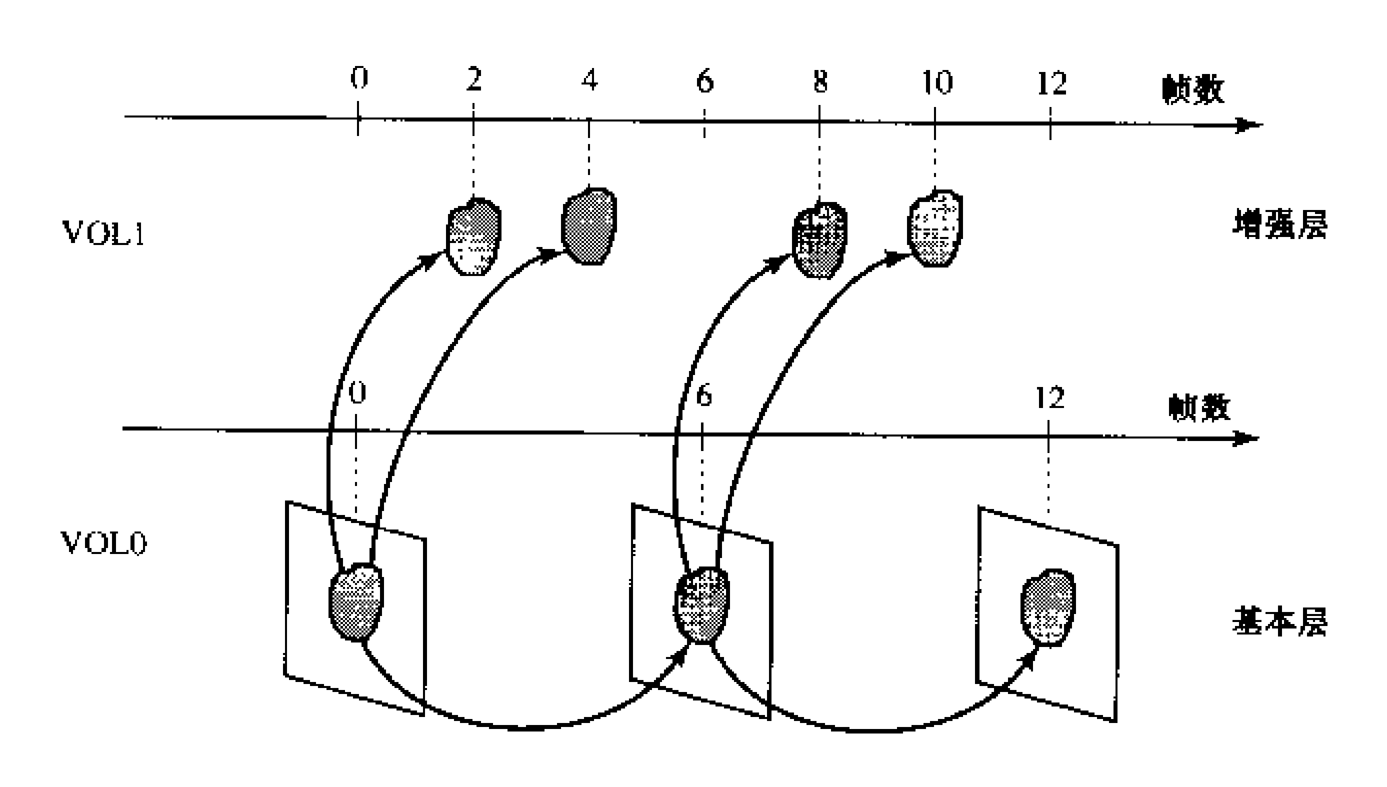


图 11.9　类型 1 的 OTS 增强结构，具有预测的视频物体平面(P-VOP)(得到MPEG-4许可)

图 11.11 示出了一个类型 2 OTS 的例子，其中 VO0 (视频物体 0) 是仅包含背景且没有可分级性层的整个帧的序列。VO1 是一个特殊物体的具有两个可分级性层 VOL0 和 VOL1 的序列。VOL1 表示与 VOL0 相同的物体，但被编码以达到高于 VOL0 的帧率。在这个例子中，

VOL0 被认为是 OTS 的基本层，VOL1 被认为是 OTS 的增强层。注意，VO0 可能不具有与其他的 VO 相同的帧率。

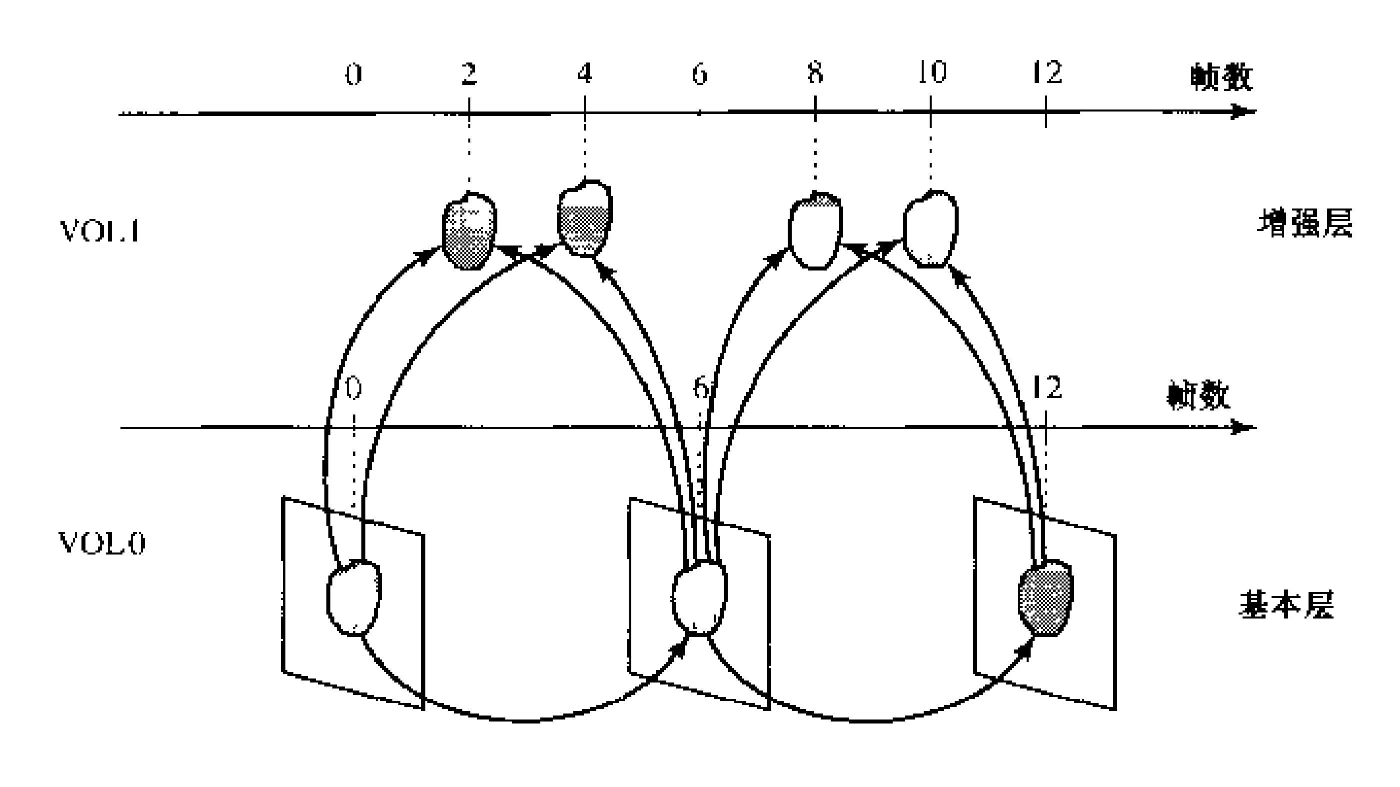


图 11.10 类型 1 的 OTS 增强结构，具有 B-VOP

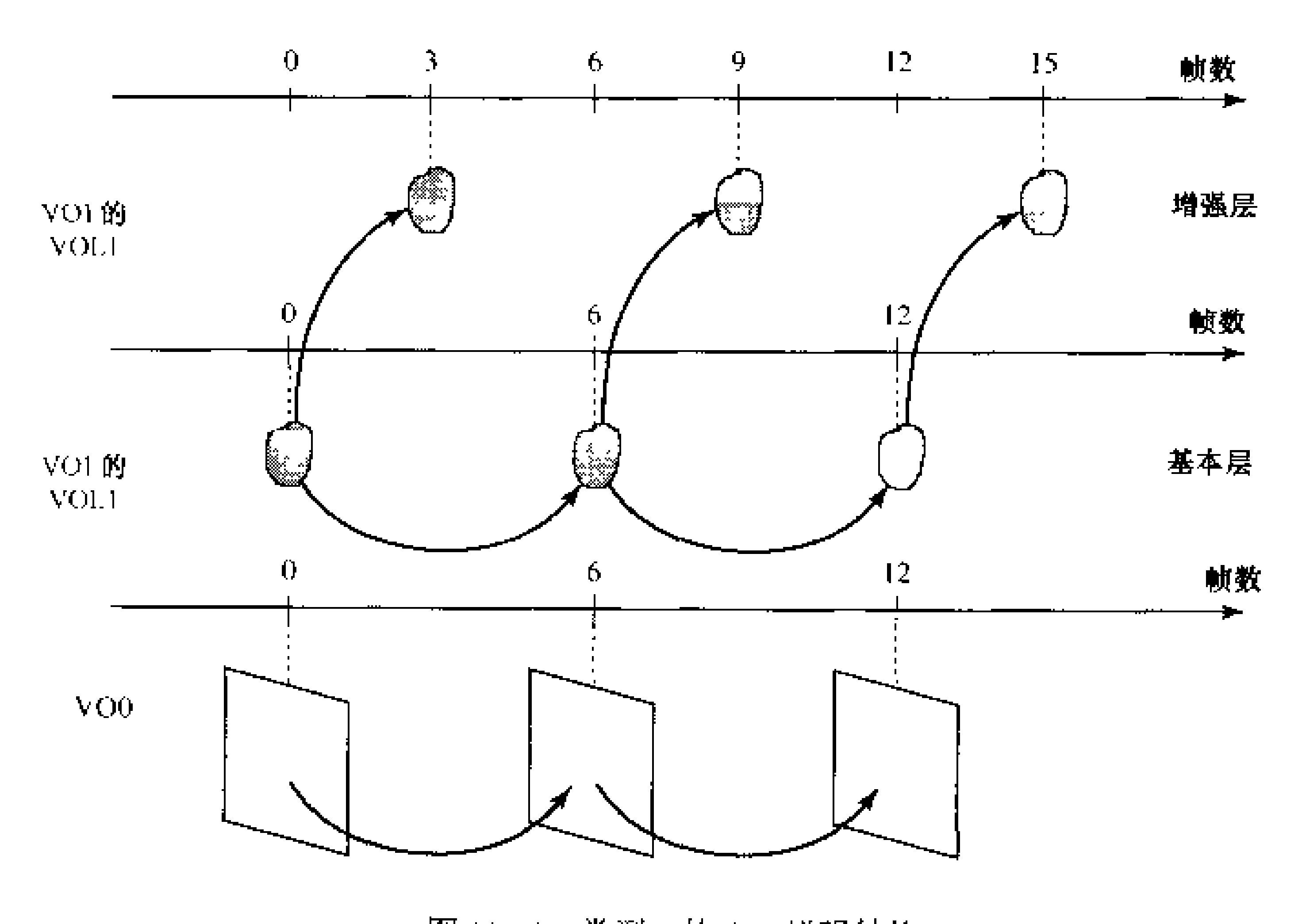


图 11.11 类型 2 的 OTS 增强结构

11.3 基于小波变换的编码

最近出现的离散小波变换(DWT)由于其在表示非平稳图像信号方面的灵活性和适应人类视觉特征的能力，已经成为图像和视频压缩方面的有力工具[16,15]。小波表示以时间和频率的局部化提供了信号的一个多分辨率/多频率表示。在图像和视频编码应用中是非常期望这种性质的。首先，真实世界的图像和视频实际上是非平稳的。小波变换把非平稳信号分解成

一系列多级子带,每个子带中的分量变得相对比较平稳,因此易于编码。同时,可使编码方案和参数适应于每个子带的统计性质,因此编码每个平稳的分量比编码整个非平稳信号的效率更高。另外,如心理物理学和生理学方面的研究结果所指出的,小波表示与早期人类视觉所体验的空间调谐的频率调制性质相匹配。最后,小波变换所提供的多分辨率/多频率分解使它本身易于适合可分级的比特流。

与基于 DCT 的方法一样,基于小波变换的图像编码包括三个步骤:(1)小波变换;(2)量化;(3)熵编码。小波变换把图像信号转换成一系列系数,这些系数对应于多分辨率和频率段上的分级空间表示。一般把小波系数组织为分层的数据结构,以便可以更有效地进行比特分配和数据压缩。量化允许以一定的失真为代价来降低码率。最后,熵编码把量化后的系数编码成一组压缩的二进制比特流。基于小波变换的编码有许多变种。小波变换一般是整体进行的,但也已经开发了块状的小波变换以提高实施效率 [9]。最近也开发了矢量小波变换和任意形状的小波变换 [12, 10]。量化可以有好几种类型:标量、矢量或格形编码 (TCQ)。熵编码可以是哈夫曼或算术编码,使用固定的或者自适应的码,根据比特平面或整个样点实现。当把小波应用于视频编码时,需要减少时间冗余的机制。已经提出了在空间和小波域中的运动补偿、三维小波变换和运动补偿的三维小波视频编码算法 [26, 21]。

对于图像编码,已经证明在压缩效率和增强特性(例如可分级性)方面,小波变换比基于 DCT 的编码技术优越得多。因此 MPEG-4 和 JPEG2000 都选择基于小波的方案作为编码静止纹理和图像的基础。然而,仍有待于证明基于小波的视频编码在压缩方面显著地优于基于 DCT 的方案。在参考文献[25]中包括了图像和视频 DCT 与小波编码的比较。大多数现有的视频编码标准(H.261,H.263,MPEG-1 和 MPEG-2)已采用在第 9 章中介绍的基于块的混合视频编码框架 [7,8,3,4]。MPEG-4[5]也采用基于 DCT 的方案编码自然图像,尽管基于小波的编码被用于静止图像和图形。目前在基于小波的视频编码研究方面非常活跃,目的在于更大地提高压缩率和可分级性。

这一节我们从描述最流行的基于小波的图像编码方案(称为基于零树的方案)开始。然后是一个运动补偿视频编解码的例子,以说明小波在视频编码中的应用。由于小波变换的基本数学公式已经广泛地包含在有关著作中 (见参考文献[23,1]),本节将集中在基于小波编码方法的压缩方面。

11.3.1 静止图像的小波编码

EZW 方法　基于零树的编码概念最早是由 Shapiro 提出的,称为嵌入式零树小波(EZW, embedded zero-tree wavelet)编码[19]。除了优越的压缩性能外,零树小波编码的优点包括简单性、嵌入式比特流结构、可分级性和精确的比特率控制。零树小波编码基于三个关键的思想:(1)用小波变换去相关;(2)利用小波变换的内在自相似性在各级之间预测重要信息的位置;(3)用自适应算术编码进行通用的无损数据压缩。

离散小波变换把输入图像分解成一系列不同分辨率的子带。最粗糙的子带称为 DC 带,是原始图像的低通近似,而其他的子带是较精细尺度的细节。在一个分层子带系统(例如小波变换系统)中,除了最低频子带外,在给定级上的每个系数都可能与下一个较精细级上的相似方位的一组系数相关。粗糙级上的系数称为父,而所有在相同空间位置和在下一个较精细级上的相似方位的系数称为子。

作为例子,图 11.12 示出了一个由三级小波分解所得的小波树。对于最低频子带,即这个例子中左上角的方块,父子关系的定义是每一个父节点具有三个子节点,每个子带一个,它们位于同样的级和空间位置,但在不同的方位上。对于其他的子带,每一个父节点具有四个子节点,它们在相同方位的下一个较精细的级上。

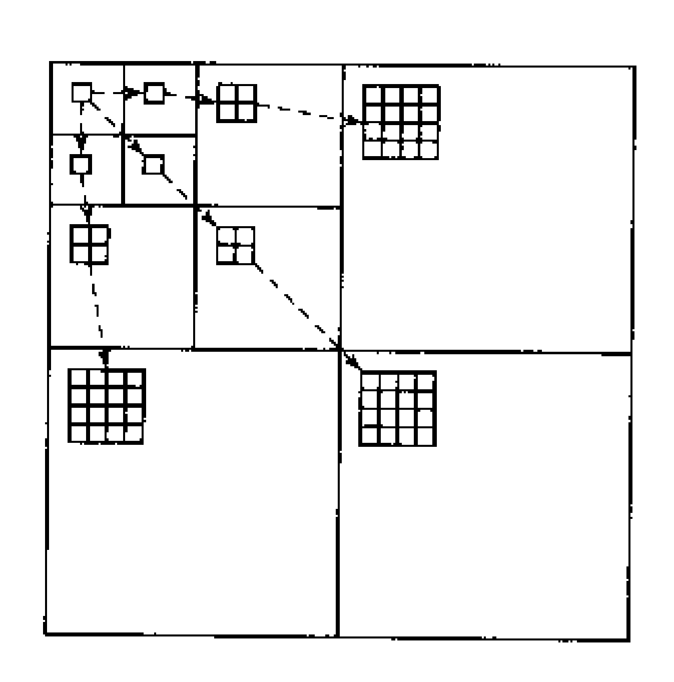


图 11.12　小波系数的父子关系

DC 子带的小波系数是与其他子带独立编码的。如图 11.13 所示,当前系数 X 由与它相邻的三个其他的量化系数(即 A, B 和 C)自适应地预测,并从当前系数减去预测值。预测值可按下式得到:

$$\hat{X} = \begin{cases} C & |A - B| < |B - C| \\ A & \text{其他} \end{cases} \tag{11.3.1}$$

如果任何一个相邻的系数(A, B 或 C)不在图像中,为了预测,应把它的值设置为零。

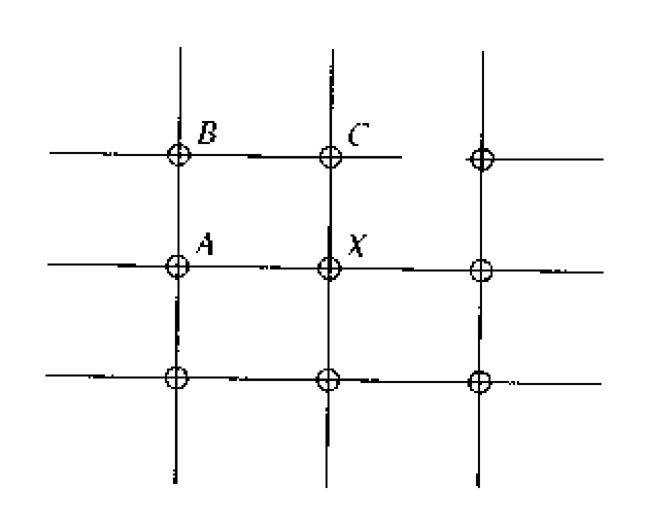


图 11.13　DC 系数的自适应预测编码

在比特流中,首先编码量化步长;然后,差分量化索引的最小值幅度(称为带偏移量)和最大值幅度(称为带最大值)被编码成比特流。参数带偏移量是负整数或零,而参数带最大值是正整数。所以只有这些参数的幅度被编码成比特流。差分量化索引用算术编码器以光栅扫描顺序被编码,从左上角的索引开始,到右下角结束。该模型由每个编码差分量化索引的比特进行更新,以适应 DC 带的统计特性。

EZW 逐子带地扫描小波系数。父节点在其任何子节点之前被扫描。每个系数都与当前阈值 T 进行比较。如果系数的幅度大于 T,它就是重要的。然后用两个符号(负重要的或正

重要的)对这样的系数进行编码。零树根符号用于表示小于 T 且它所有的子节点也小于 T 的系数。孤零符号表示小于 T 但至少有一个子节点不小于 T 的系数。对于重要的系数,EZW 进一步用一个连续逼近的量化方案编码系数的值。编码是通过比特平面进行的,导致一个嵌入式比特流。

ZTE 和 MZTE 方法　近年来 EZW 已经被显著地改进。例如 SPIHT[18]、预测 EZW[13]、零树熵(ZTE)编码[17]和更一般情况的多级 ZTE(MZTE)[20]。使用零树的形状自适应小波编码也已经被提出,用于编码具有任意形状的物体[10]。

零树熵编码是基于 EZW 编码的,但与 EZW 编码有很大的不同。与 EZW 类似,ZTE 编码利用图像小波变换内在的自相似性来预测小波各级之间信息的位置。尽管 ZTE 不生成完全嵌入式的比特流(如 EZW),但它获利于灵活性以及其他胜于 EZW 编码的优点,包括在编码效率、简单性和空间可分级性方面实质性的改善。ZTE 编码是通过把零树符号分配给系数,然后对系数值连同它的符号以两种不同的扫描顺序之一进行编码。在 ZTE 中用的四个零树符号是零树根(ZTR)、估价的零树根(VZTR)、估价(VAL)和孤零(IZ)。然后零树符号和量化系数用一个具有给定的符号字母表的自适应算术编码器进行无损编码。算术编码自适应地跟踪零树符号的统计特性,并用三种模式来编码值:(1)类型,为了编码零树符号;(2)幅度,为了编码比特平面方式的值;(3)符号,为了编码值的符号。对于每个系数,首先编码它的零树符号;如果有必要的话,再编码它的值。值的编码分为两步。首先,用适当的概率模式以比特平面方式编码它的绝对值;然后,用一种二进制概率模型编码符号。

MZTE 编码技术基于 ZTE 编码,但使用一种新的框架改善 ZTE,并把它扩展为一种完全可分级且仍然非常有效的编码技术。在第一个可分级层,用与 ZTE 编码相同的方式生成零树符号,并把零树符号与该可分级层的非零小波系数一起进行编码。对于下一个可分级层,零树映射与相应值的细节一起进行更新。在每个可分级层中,只有当一个系数在前一个可分级层中作为 ZTR 或 IZ 编码时,才为它编码一个新的零树符号。如果这个系数在前一层中是作为 VZTR 或 VAL 编码的,那么在当前层仅编码它的细节值。一个对于残差的附加概率模型用于编码在任何前面的可分级层中用 VAL 或 VZTR 符号编码的系数的细节。进一步地,为了利用可分级层之间高度相关的零树符号,基于 MZTE 的前一个可分级性层中的系数的零树符号,使用上下文模型以更好地估计零树符号的分布。

图 11.14(a)和(b)中的图像是分别用 JPEG 和 MZTE 压缩方案,在相同的 45:1 的压缩比下获得的。结果表明 MZTE 方案产生比 JPEG 好得多的图像质量,很好地保留了精细的纹理区域,而且没有块效应。在参考文献[25]中的研究表明,基于小波的编码器在比特率相同的情况下,在 PSNR 方面一般优于基于 DCT 的编码器大约 1 dB。

使用 MZTE 方法,我们可以很容易获得空间和质量可分级性。通过从 MZTE 压缩图像的比特流中解码越来越多的比特,可获得越来越大和越来越清晰的原始图像的再现。

形状自适应 DWT　基于小波的编码可扩展到任意形状的物体[10]。通过形状自适应 DWT (SA-DWT),物体的形状掩膜被分解成子带金字塔,从而我们知道哪些小波树的节点具有有效的小波系数而哪些具有“不必在意”的值。我们必须注意只有这些不必在意的值(对应边界外的像素或外节点)的多分辨率任意形状物体的编码方式。各种基于小波的编码方法,包括 EZW 和 ZTE 可以扩展到 SA-DWT。在本小节中,我们讨论如何把 ZTE 方法扩展到形状自适应的情况。

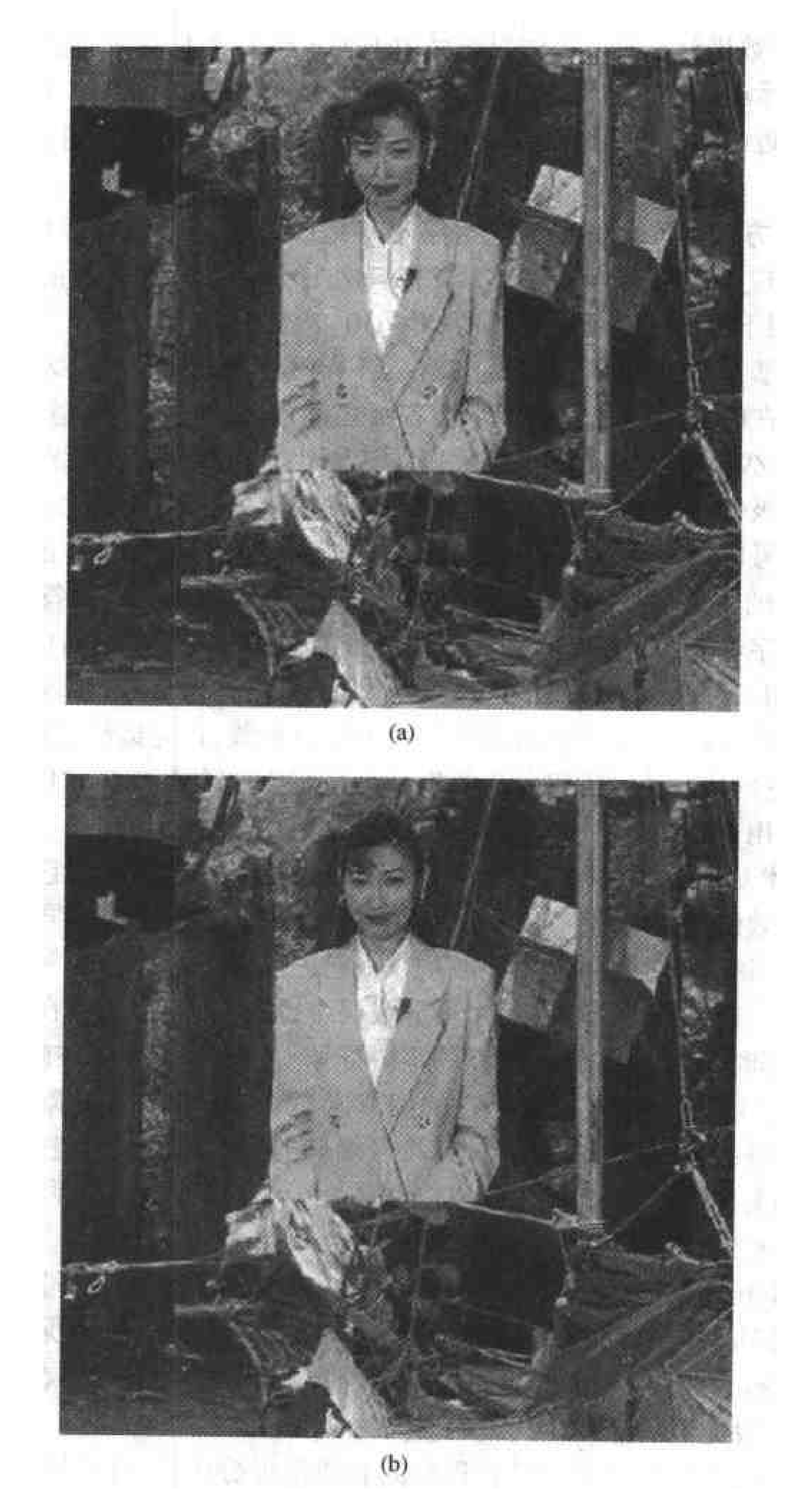


图 11.14 (a)用 JPEG 基线方法编码的图像(Y,U 和 V 分量的 PSNR 分别是 28.36,34.74,34.98 dB);(b)用MZTE方法编码的图像(Y,U和V分量的PSNR分别是30.98,41.68,40.14 dB)。(a)和(b)的压缩比都是45:1。只示出了Y分量部分

如在参考文献[10]中所讨论的,SA-ZTE 把图像域的任意形状的物体分解为一个分层的结构,具有一系列不同分辨率的子带。每个子带具有一个与它相关的对应的形状掩膜,用以确定该子带中有效系数的位置。在树中有三种类型的节点:零、非零和外节点(具有不必在意的

值)。任务是要把零树编码方法扩展到具有外节点的情况。简单的方法是把那些不必在意的值设置为零,然后应用零树编码方法。然而,这需要比特以"不必在意的树"的形式(其中父节点和它的所有子节点都具有不必在意的值)编码外节点。这是浪费比特的,因为外节点不需要编码,形状掩膜已经指明了它们的状态。因此,我们对待外节点应该与对待零不同。尽管我们不想使用比特来编码外节点,但必须决定怎样处理它的子节点。一种方法是关于不必在意节点的子节点的状态不编码任何信息。这样,我们总是假设它有四个子节点要进一步检查。当解码器扫描到这个节点时,形状信息就会告知它这是一个不必在意的节点,且它将继续扫描这个不必在意节点的四个子节点。如此进行,就不必对树结构中所有不必在意的节点编码。当树结构中仅有稀疏的有效节点时,这种方法工作得很好。这种方法的一个缺点是,即使不必在意节点具有四个零树根子节点,如果把不必在意节点作为零对待,它仍然必须编码四个零树根符号而不是编码一个。另一种方法是有选择地把外节点作为零对待。这等价于生成另一种符号来编码某些不必在意的值。通过广泛的试验,已经发现不编码外节点的方法总的来说工作得较好。这种编码算法的详细描述可在参考文献[10]中找到。

已经对 SA-DWT 编码技术进行了广泛的试验,并且其结果与 SA-DCT 编码进行了比较。物体的形状用 MPEG-4 的形状编码工具进行编码。图 11.15 给出了由 SA-DCT 和 SA-ZTE 方法获得的 PSNR 与比特率的关系曲线。形状比特不包括在比特率内,因为它们不依赖于纹理编码方案。只有纹理比特率用于压缩。比特率(以 bpp 为单位)是基于具有重建形状的物体中的像素数计算的,并且 PSNR 值也是在重建形状中的像素上计算的。很清楚,SA-ZTE 编码可以比 SA-DCT 达到更好的编码效率,具有 1.5~2 dB 的 PSNR 改善。图 11.16(a)和(b)分别示出了由 SA-DCT 和 SA-ZTE 编码重建的物体。

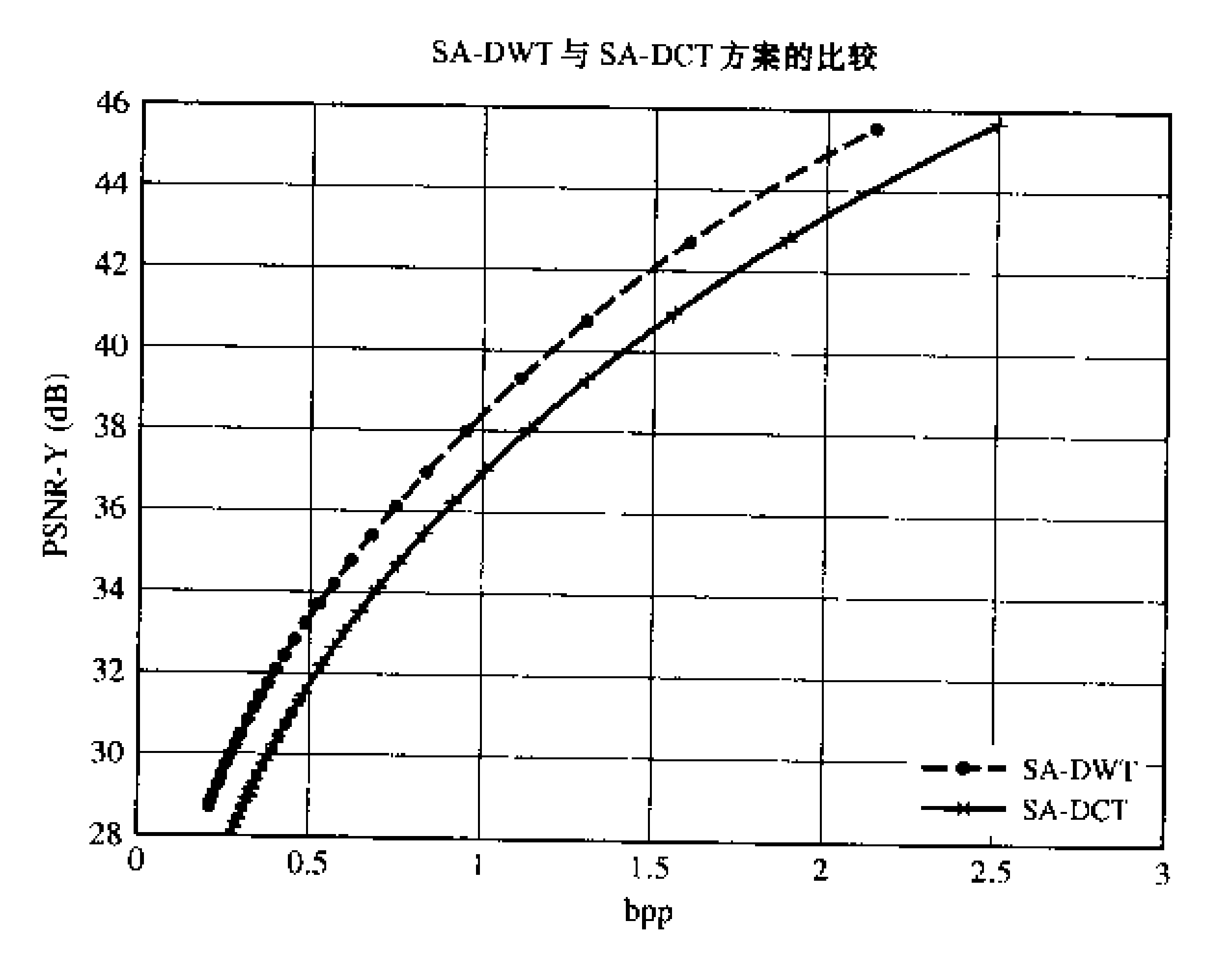


图 11.15 SA-DWT(在这种情况下是 SA-ZTE)编码与 SA-DCT 的比较

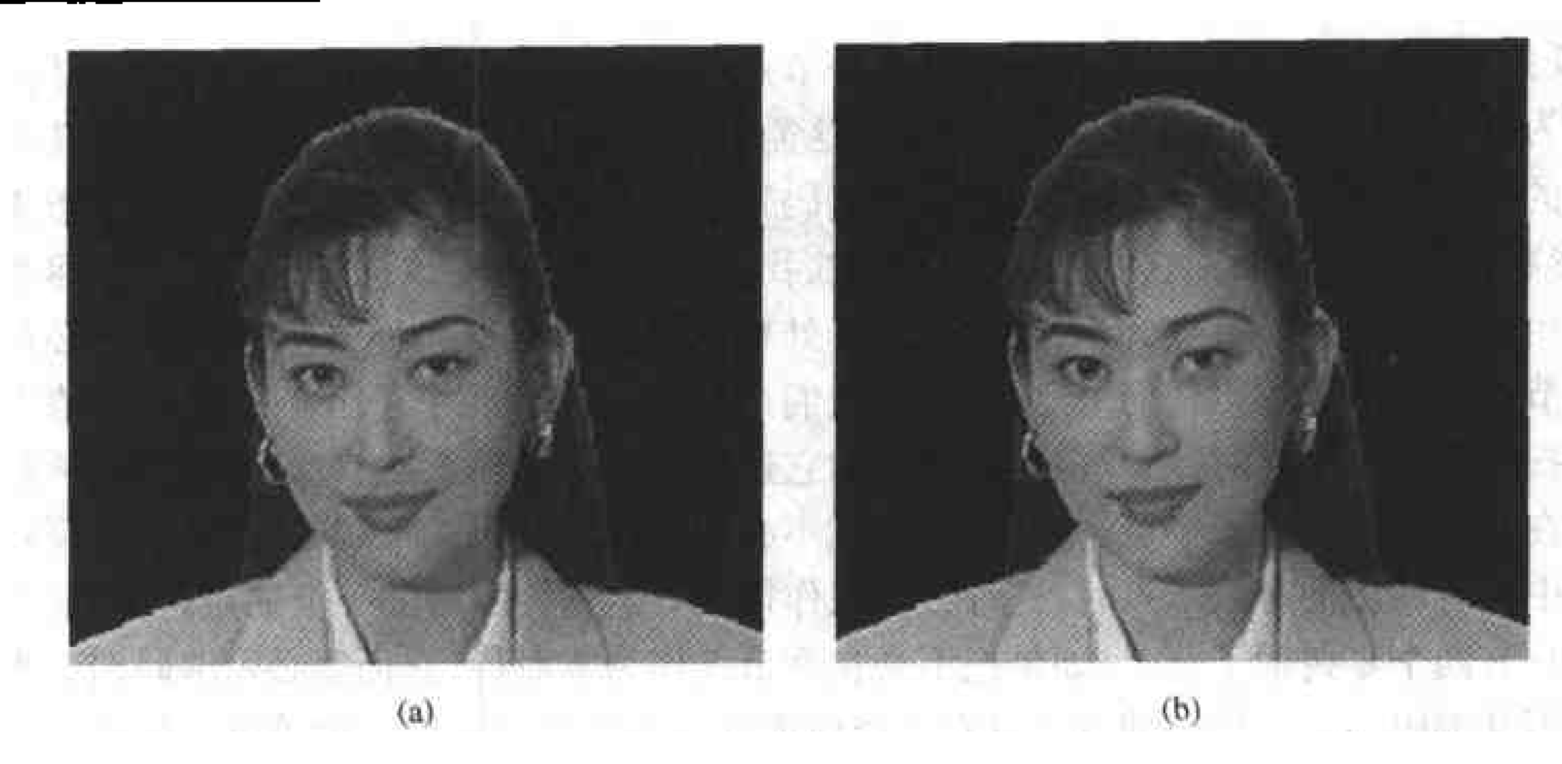


图 11.16 用 SA-DCT 和 SA-ZTE 重建的物体：(a)SA-DCT(1.0042 bpp; PSNR-Y = 37.09 dB; PSNR-U = 42.14 dB; PSNR-V = 42.36 dB)；(b)SA-ZTE(0.9538 bpp; PSNR-Y = 38.06 dB; PSNR-U = 43.43 dB; PSNR-V = 43.25 dB)。仅示出了Y分量的部分

11.3.2 视频的小波编码

视频的小波编码技术可以分为三类：(1)空间域运动补偿后跟二维小波变换；(2)小波变换后跟频率域运动补偿；(3)具有或不具有运动估计的三维小波变换。不同的运动估计算法、量化方案和熵编码方法可应用于这三类中的每一类。

我们用一个视频编解码器的例子来说明基于小波的视频编码过程[17]。这个编解码器由五个主要部分组成：(1)三级的运动估计和补偿；(2)自适应小波变换；(3)多级量化；(4)零树熵编码；(5)智能比特率控制。编解码器方框图如图 11.17 所示。首先，进行三级运动估计和补偿过程，包括全局运动估计、可变的块尺寸运动估计和重叠块运动补偿。然后用小波滤波器把运动补偿的残差帧分解成一系列多分辨率金字塔子带，后接一个具有不同步长和死区的多级量化器。最后用一个类似 ZTE 的(但对残差视频小波系数最优化的)编码器对量化系数进行熵编码。

全局运动估计用于估计摄像机运动的影响并把图像序列分裂成不同的层。每一层具有它自己的全局和局部运动估计。全局运动是基于二维仿射运动模型估计的。而且它是在局部运动估计之前进行的。

全局运动估计后，用四叉树结构把帧分裂为可变尺寸的块，范围从 8×8 到 256×256 个像素。然后用传统的块匹配为各个块寻找 MV。OBMC(见 9.3.2 节)用被各自的 MV 位移的相邻块的加权平均作为当前块的预测。它产生一个较平滑的运动补偿残差帧的版本。由于可变块尺寸用于运动估计，OBMC 使用最小的块尺寸(在这种情况下是 8×8)以及连带的窗函数作为用于预测的重叠单元。

小波变换用于运动补偿的残差帧。由于对残余误差帧不同子带上的能量分布倾向于均匀，所以选择小波滤波器的常规准则(紧缩的支撑区)变得不太重要。这里更重要的准则是小波滤波器的时间局部化。具有好的时间局部化的小波滤波器倾向于局部化在残差帧(比在原始图像)中更经常遇到的边缘。这表明短抽头滤波器(具有较好的时间局部化但较差的紧缩支撑区)比长抽头滤波器(具有较好的紧缩支撑区但较差的时间分辨率)的效果更好。确实，已经发现哈尔(Haar)变换[1]能相当好地进行这个工作。

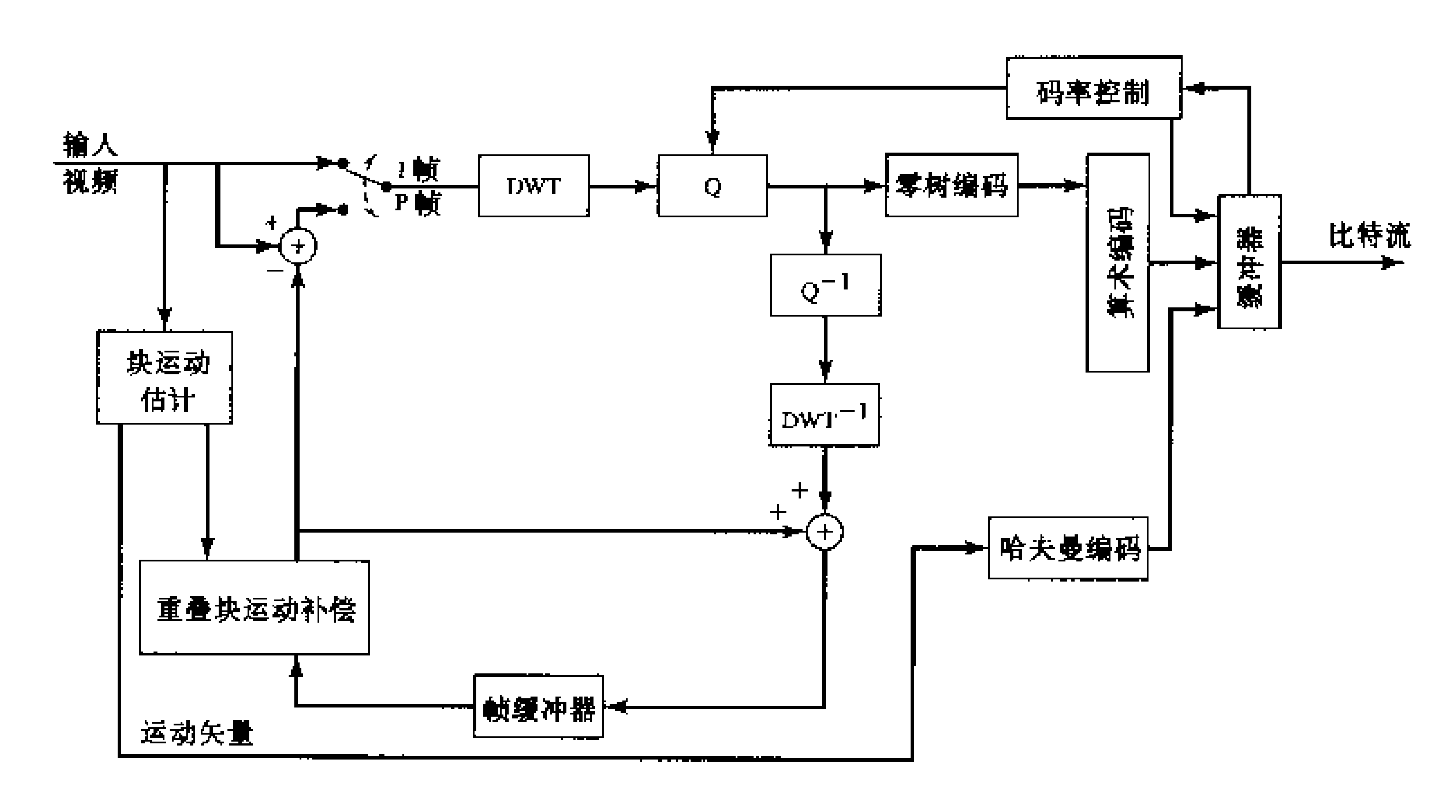


图 11.17　基于小波的视频编解码器的方框图

DWT 后接小波变换系数的标量量化，然后进行 ZTE。现有编码器中的比特分配和码率控制可分为三级：(1)具有可变帧率的时间域比特分配；(2)帧间比特分配，给出最佳的帧间比特分配；(3)帧内比特分配，在帧内产生最佳的比特分配。

已把这种方法与 MPEG-4 中基于 DCT 的编码器进行了比较[6]。在 PSNR 方面，两种方法产生相似的性能[25]。已经发现编码系统中量化的最佳化、熵编码以及各部分之间复杂的相互作用比最优化变换本身更重要。

11.4　小结

- 可分级性是编码方法中的一个非常有用的特性，因为它允许灵活性，并自适应于：(1)变化的网络带宽；(2)变化的信道条件；(3)不同的计算能力。对于在具有时变带宽或误差特征网络上的传输，或传输到具有不同接入速率或处理能力的接收机，可分级编码器是所期望的。对于同一个视频文件可以被具有不同的通信链路和计算能力的用户访问的视频流应用系统，可分级性也是所期望的(在第 15 章中将进一步讨论视频流)。
- 可分级编码器可以具有粗间隔度(两层或三层，这些也称为分层编码器)或细间隔度。在细间隔度的极限情况下，比特流是完全嵌入式的，并可在任何点截断。
- 质量、空间、时间和频率可分级性是基本的可分级机制。可以把这些方法组合起来以实现较细间隔度。
- 在固定的空间时间分辨率下获得嵌入式流的一种方法是相继地编码比特平面，如在 MPEG-4 标准的细间隔度可分级性模式中和在大多数基于小波的编码器中所用到的。
- 可以很容易地使基于小波的编码器生成嵌入式流，因为小波变换提供信号的多级表示。

11.5　习题

11.1　分层编码器与嵌入式编码器之间有什么区别？给出每类编码器的例子。

11.2 描述四个基本的可分级性模式:质量、空间、时间和频率。

11.3 11.1.6节中所述的FGS方法是如何工作的?FGS和质量可分级性之间有何区别?

11.4 考虑i.i.d.高斯过程。假设我们可以设计一个达到RD界限的一层编码器;也就是说,失真与码率的关系是 $D=\sigma_x^2 2^{-2R}$,其中 σ_x^2 是信源信号的方差。说明两个量化层的质量可分级编码器将达到与这个一层的编码器相同的RD界限。把证明扩展到任意的层数。(这个结果意味着对于高斯过程,SNR可分级性能达到分层功能,而不牺牲编码效率;这就是所谓的高斯过程的连续细化性质。为进一步研究这一课题,见参考文献[2]。)

11.5 以在习题9.15中为基于块的混合视频编码器所编的程序开始,为实现两层质量可分级性生成代码。比较原始编码器与分层编码器对一个测试序列的编码效率。

11.6 以与习题11.5相同的程序开始,为实现两层空间可分级性生成代码。比较原始编码器与分层编码器对一个测试序列的编码效率。

11.7 块状的变换(例如DCT)与小波变换之间的主要区别是什么?每一个的优缺点是什么?

11.8 如何向想知道EZW方法是如何工作的同学描述此方法?

11.9 为什么小波变换更适于生成嵌入式流?如何分类由EZW方法所提供的可分级性的类型?它提供质量、空间和/或频率可分级性吗?

11.10 用哈尔小波滤波器(低通滤波器:[1,1],高通滤波器[1,-1])实现图像的两级小波变换。当一个所选择的子带中的系数置为零时,考查所产生的图像的质量。基于这些图像,评论小波系数的一个特殊的带揭示了哪类特征。

11.6 文献目录

[1] Arkansu, A., and R. A. Haddad. *Multiresolution Signal Decomposition: Transforms, Subbands, Wavelets*. London: Academic Press, 1996.

[2] Equitz, W. H. R., and T. M. Cover. Successive refinement of information. *IEEE Trans. Inform. Theory* (March 1991), 37:269–75.

[3] ISO/IEC. IS 11172: Information technology—Coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbps, 1993. (MPEG-1).

[4] ISO/IEC. IS 13818-2: Information technology—Generic coding of moving pictures and associated audio information: Video, 1995. (MPEG-2 Video).

[5] ISO/IEC. IS 14496: Information technology—coding of audio-visual objects—part 2: Visual, 1999. (MPEG-4 Video).

[6] ISO/IEC. Very low bitrate video codec. Public document, ISO/IEC MPEG96/M0637, Munich, January 1996. (MPEG-4 VM).

[7] ITU-T. Recommendation H.261: Video codec for audiovisual services at px64 kbits, 1993.

[8] ITU-T. Recommendation H.263: Video coding for low bit rate communication, 1998.

[9] ISO/IEC JTC1/SC29/WG1. FCD 15444-1: Information technology—JPEG2000 image coding system, March 2000. (Available at http://www.jpeg.org/FCD15444-1.htm.)

[10] Li, S., and W. Li. Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding. *IEEE Trans. Circuits Syst. for Video Technology* (Aug. 2000), 10:725–43.

[11] Li, W. Overview of fine granularity scalability in MPEG-4 video standard. *IEEE Trans. Circuits Syst. for Video Technology* (March 2001), 11(3):301–17.

[12] Li, W., and Y.-Q. Zhang. Vector-based signal processing and quantization for image and video compression. *Proceedings of the IEEE* (Feb. 1995), 671–91.

[13] Liang, J. Highly scalable image coding for multimedia applications. *Fifth ACM International Multimedia Conf.* Seattle, WA, (Nov. 1997), 11–19.

[14] Ling, F., W. Li, and H. Q. Sun. Bitplane coding of DCT coefficients for image and video compression. *SPIE Conf. Visual Commun. Image Processing* (Jan. 1999).

[15] Mallat, S. Multifrequency channel decompositions of images and wavelet models. *IEEE Trans. Acoust., Speech, Signal Processing* (Dec. 1989), 37(12):2091–2110.

[16] Mallat, S. A theory of multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Analysis and Machine Intell.* (July 1989), 11(7):674–93.

[17] Martucci, S. A., I. Sodagar, T. Chiang, and Y.-Q. Zhang. A zerotree wavelet video coder. *IEEE Trans. Circuits Syst. for Video Technology* (Feb. 1997), 7(1):109–18.

[18] Said, A., and W. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circuits Syst. for Video Technology* (June 1996), 6:243–50.

[19] Shapiro, J. M. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. Signal Processing* (Dec. 1993), 41:3445–62.

[20] Sodagar, I., H.-J. Lee, P. Hatrack, and Y.-Q. Zhang. Scalable wavelet coding for synthetic/natural hybrid images. *IEEE Trans. Circuits Syst. for Video Technology* (March 1999), 9:244–54.

[21] Taubman, D., and A. Zakhor. Multirate 3-D subband coding of video. *IEEE Trans. Image Process.* (Sept. 1994), 3:572–88.

[22] van der Schaar, M., and H. Radha. A hybrid temporal-SNR fine-granular scalability for Internet video. *IEEE Trans. Circuits Syst. for Video Technology* (March 2001), 11(3):318–31.

[23] Vetterli, M., and J. Kovacevic. *Wavelets and Subband Coding*. Englewood Cliffs, NJ: Prentice Hall PTR, 1995.

[24] Wu, F., S. Li, and Y.-Q. Zhang. A framework for efficient progressive fine granularity scalable video coding. *IEEE Trans. Circuits Syst. for Video Technology* (March 2001), 11(3):332–44.

[25] Xiong, Z., K. Ramachandran, M. Orchard, and Y.-Q. Zhang. A comparative study of DCT- and wavelet-based image coding. *IEEE Trans. Circuits Syst. for Video Technology* (Aug. 1999), 9(5):692–95.

[26] Zhang, Y.-Q., and S. Zafar. Motion-compensated wavelet transform coding for color video compression. *IEEE Trans. Circuits Syst. for Video Technology* (Sept. 1992), 2(3):285–96.

第12章　立体和多视序列处理

体视序列和(更一般情况下)多视序列的采集和显示有许多应用。在工业中,立体显示用于机器人在危险环境下进行物体的处理和检查。对于医学应用场合,体视探测提供比单视探测更有用的信息。对于科学勘探和显像,体视或多视显示能使观看者通过平面的二维屏幕体验三维环境。这种应用已经存在很久了,传统上立体序列是以模拟的形式进行采集、存储和显示的。然而,近年来数字电视技术的发展推进了三维视觉。随着数字电视(包括 HDTV)变成现实,可以预见数字三维电视将是电视工业的下一个“量子跳跃”,除了有希望使图像更清晰外,还提供三维感觉。另一个较近期的立体/三维显示的应用是虚拟现实和人机对话,这时三维表演和显示给观看者一种在遥远的(或不存在的)地方与人或物体在物理上相互作用的幻觉。

在本章中,我们从简单地讨论人类视觉系统感觉深度的机理开始(12.1 节)。然后转到立体成像的技术方面,说明在立体图像对中成像点的三维位置与它的投影的关系(12.2 节)。接下来我们讨论在立体序列处理中技术上最有挑战性的问题:位差和由此引起的深度估计(12.3 节)。然后描述从已存在的视图内插中间视图的方法(12.4 节)。最后,我们给出编码体视和多视视频的方法(12.5 节)。选择这些题目,是因为它们属于立体和多视系统的信号处理领域;关于立体和多视采集与显示系统,可参见参考文献[16]。

12.1　深度感觉

体视成像和显示受到了人类通过两只分开的眼睛来感觉深度的启发。尽管存在几种人脑用来区分不同深度物体的单目线索(例如,通过相对的大小和运动位差),但最重要和有效的机制是通过称为立体感的双目线索。为了理解体视系统是如何工作的,以及它为什么能这样工作,理解这种现象是至关重要的。在本节中,我们首先描述立体感是如何产生的。然后给出深度感觉的视觉灵敏度,这对于设计三维视频系统非常重要。

12.1.1　双目线索——立体感

为了解释立体感,我们取参考文献[17]中的一个例子。试做一下这个实验:把一个手指放在你的脸前。当看这个手指时,你的眼睛聚焦在手指上。这是通过眼睛的肌肉运动完成的,使手指的图像落在每个眼睛视网膜的中央凹处——也就是每个视网膜的中心部分。如果你继续把眼睛聚焦在手指上,同时注意背景,你将觉察到背景看起来是双重的。另一方面,如果你试图聚焦在背景上,你的手指将出现两个。这种现象示于图 12.1。

假设我们可以把左视网膜和右视网膜上的图像取出来并把它们叠在一起,我们将看到两个几乎重叠的图像。视网膜位差是指重叠的视网膜图像上相应的左和右图像点之间的水平距离。位差为零的那些点在眼睛聚焦的地方。把两个不同感觉的视图综合成一幅图像的心理能力称为融合,产生的深度感觉称为立体感。

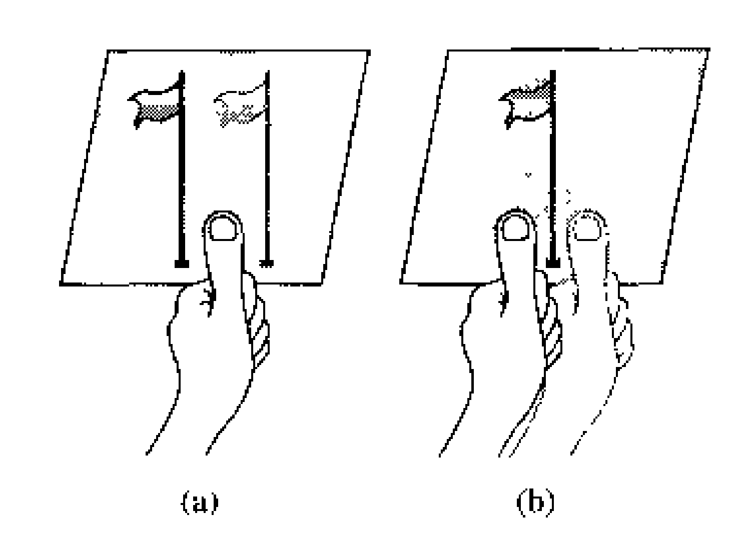


图12.1　一个理解立体感的实验：(a)当眼睛聚焦于大拇指时，背景看起来像是两个图像；(b)当眼睛聚焦于背景时，看到前景中有两个拇指

12.1.2　深度感觉的视觉敏感度阈值

我们已经知道了人是如何感觉深度信息的，接下来的问题是人的视觉系统对空间和时间上深度变化的敏感程度如何。在2.4节中，我们讨论了在单视图像中检测空间(在同一个二维平面)和时间变化的视觉阈值。我们知道作为空间频率函数的对比敏感度在低时间频率上显示带通特征(最大敏感度近似为4 cpd)，在高时间频率上显示低通特性。类似地，关于时间变化的视觉敏感度在低空间频率是带通的(最大敏感度近似于8～15 Hz)，在高空间频率是低通的。这些发现指导了视频显示系统的设计。为了显示眼睛所能看到的一切信息，设备应该能生成至少64 cpd的空间频率(以像素为单位的分辨率取决于图像的高和宽以及观看距离)和70 Hz的时间频率。

为了设计三维视频采集和显示系统，重要的问题是深度感觉中的视觉阈值。因为深度是通过视网膜位差感觉的，所以可根据位差的变化来估计深度的视觉敏感度。以下引用的结果来自Pastoor [11]。

对空间和时间上深度变化的敏感度　已经发现人眼对位差的空间和时间变化的视觉敏感度显著地低于对图像对比度变化的敏感度。因此，满足对比度敏感度函数的空间和时间带宽要求的显示系统也将能在空间和时间上满足深度变化的感觉。位差场的空间频率视觉敏感度具有带通特性，带宽范围为3～4 cpd，最大敏感度在0.2～0.5 cpd范围之间。当沿深度运动时仍可察觉位差的时间变化的上限在4～6 Hz范围内。

对位差量化的敏感度　已经发现体视敏锐度——即分辨深度刺激的能力——的阈值大约是2弧秒①。当左和右图像的水平图像分辨率至少是24 cpd时，就可感觉到这样细微的深度上的差别。对于数字显示，投影位差被量化。当量化太粗糙时，场景上的物体可能看起来像是属于不同深度上的几层平面景物，且物体沿深度的运动也可能看起来是不平稳的。

对视角量化的敏感度　在自然视觉中，当观看者移动时，他们的视点连续地运动。在用三维显示系统提供多视点时，仅可实现有限数量的视点，可以把它看做视角方向的量化。这种量化的效应是，当观察者在显示器前改变他们的位置时，在感觉的视图之间会发生突然的移动。已经证实，对于CCIR质量评价尺度主观质量被认为“好”来说，视角移动必须处于近似1弧

① 1弧秒等于1/3600度。

分[①] 的阈值以内。

不对称空间分辨率要求 各种试验已表明,在立体显示中,当一幅图像至少在一个短的期间内以视觉的空间-时间阈值所规定的足够的分辨率显示时,可以相当大地降低另一幅图像的分辨率(直到一半),而不会引起清晰度主观印象的下降。我们还不知道当长时间观看时,这样的显示会引起的视觉疲劳程度。这种不对称性质已经在立体序列压缩中进行了研究(见12.5.3节)。通过在水平和垂直方向上把一个视图的采样分辨率都减少一半,立即可以实现4倍的压缩。

双目视图的异步融合 在位差值的一定界限内,视觉系统能够融合不同步显示的一个立体对的两幅图像,只要显示之间的延迟不超过 50 ms [14]。这一实事在场连续显示系统中已得到了利用。当被显示的物体运动时,左视图和右视图之间的延迟可能引起深度失真。已经发现 160 ms 或更长的延迟会产生可视的深度失真。

12.2 立体成像原理

体视成像和显示系统是模拟人类的立体感觉设计的:用两台稍微移动了位置的摄像机采集三维场景,然后把分离的图像呈现给每只眼睛。在本节中,我们描述体视成像的原理,并说明如何从三维点在两个不同成像平面上的图像位置推出它的深度。我们将看到,事实上,尽管平行排列会导致深度与图像在数学上的简单对应关系,为了采集深度信息,两台摄像机不必一定是平行的(像人眼一样)。我们将从考虑具有任意位置的两台摄像机系统开始。然后集中于两种特殊情况:一种情况是两台摄像机共享一个相同的成像平面,另一种情况是两台摄像机是相向的。

12.2.1 任意摄像机配置

我们以考虑两台任意放置的摄像机开始。考虑任意摄像机配置,如图 12.2 所示。令$[\mathbf{R}]_l$ 和 $\mathbf{t}_l$(以及$[\mathbf{R}]_r$ 和 $\mathbf{t}_r$)表示对齐一个所选择的世界坐标($\mathbf{C}_w$)和左(和右)摄像机坐标($\mathbf{C}_l$ 和 $\mathbf{C}_r$)所要求的旋转矩阵和平移矢量。则左、右摄像机坐标 $\mathbf{X}_l=[X_l,Y_l,Z_l]^T$ 和 $\mathbf{X}_r=[X_r,Y_r,Z_r]^T$ 与世界坐标 $\mathbf{X}=[X,Y,Z]^T$ 的关系为:

$$\mathbf{X}_l=[\mathbf{R}]_l\mathbf{X}+\mathbf{T}_l;\quad \mathbf{X}_r=[\mathbf{R}]_r\mathbf{X}+\mathbf{T}_r \tag{12.2.1}$$

由定义,矩阵$[\mathbf{R}]_l$ 和$[\mathbf{R}]_r$ 是标准正交的。从公式(12.2.1)消去 $\mathbf{X}$ 得到:

$$\mathbf{X}_r=[\mathbf{R}]_r[\mathbf{R}]_l^T(\mathbf{X}_l-\mathbf{T}_{rl})+\mathbf{T}_{rl}=[\mathbf{R}]_{rl}\mathbf{X}_l+\mathbf{T}_{rl} \tag{12.2.2}$$

其中

$$[\mathbf{R}]_{rl}=[\mathbf{R}]_r[\mathbf{R}]_l^T,\quad \mathbf{T}_{rl}=\mathbf{T}_r-[\mathbf{R}]_r[\mathbf{R}]_l^T\mathbf{T}_l \tag{12.2.3}$$

上述公式描述左摄像机和右摄像机坐标之间的关系。

回想当世界坐标与摄像机的坐标一致时,三维点的世界坐标 $\mathbf{X}=[X,Y,Z]^T$ 与图像坐标 $\mathbf{x}=[x,y]^T$ 的关系为公式(5.1.2)。分别应用这种关系于 $\mathbf{X}_l$ 和 $\mathbf{X}_r$,得到左和右图像坐标 $\mathbf{x}_l=[x_l,y_l]^T$ 和 $\mathbf{x}_r=[x_r,y_r]^T$:

---

① 1 弧分等于 1/60 度。

$$x_l = F_l \frac{X_l}{Z_l}, \quad y_l = F_l \frac{Y_l}{Z_l}$$
$$x_r = F_r \frac{X_r}{Z_r}, \quad y_r = F_r \frac{Y_r}{Z_r} \tag{12.2.4}$$

把这种关系代入(12.2.2)得到:

$$\frac{Z_r}{F_r}\begin{bmatrix} x_r \\ y_r \\ F_r \end{bmatrix} = \frac{Z_l}{F_l}[\mathbf{R}]_{rl}\begin{bmatrix} x_l \\ y_l \\ F_l \end{bmatrix} + \mathbf{T}_{rl} \tag{12.2.5}$$

应用上述关系,我们可以确定任何三维点的 Z_r 和 Z_l,只要它的图像位置 $\mathbf{x}_l$ 和 $\mathbf{x}_r$ 为已知。尽管有三个公式,但仅有两个是独立的,从它们可得到惟一的解。一旦知道 Z_r 和 Z_l,我们就可用公式(12.2.4)确定 X_l, X_r, Y_l, Y_r,最后用公式(12.2.1)导出参考坐标(X,Y,Z)。这就是由立体图像确定三维物体的结构(即它的各个点的三维位置)的基本原理。这个过程通常称为立体三角测量。

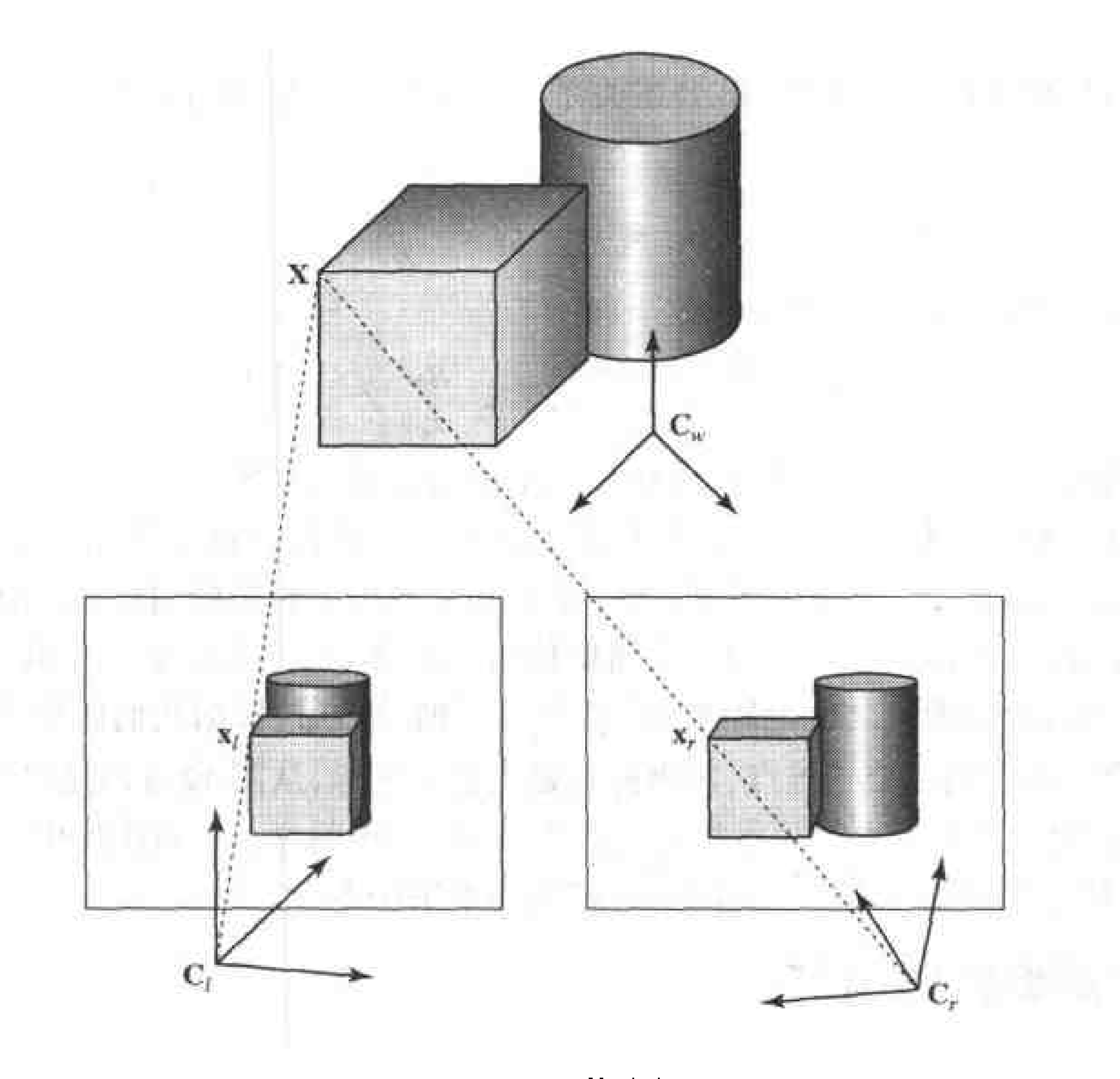


图 12.2 一个立体成像系统

重建的三维点(X,Y,Z)实际上是来自左图像点和右图像点的投影线的交点。如果两个图像点不正好是一个三维点的投影,则它们的投影线不会相交。这将使公式(12.2.5)矛盾。最小平方解将是连接这两条投影线的最短线的中点[22]。

相同三维位置的左、右图像坐标之间的位移矢量称为位差矢量(DV)或简称位差[1]。位差矢量可以在左或右图像坐标上定义。例如,如果我们用右图像位置作为基准,那么位差定义为左坐标关于给定的右坐标的位移;即 $\mathbf{d}(\mathbf{x}_r)=\mathbf{x}_l(\mathbf{x}_r)-\mathbf{x}_r$[2]。由立体图像进行三维结构估计的主要困难在于确定位差矢量;或等价地是建立左、右图像点之间的对应。

12.2.2　平行摄像机配置

立体成像最流行的配置是使用具有平行成像平面的两台摄像机,它们位于世界坐标的相同 X-Y 平面,如图 12.3 所示。两台摄像机之间的距离称为基线距离,用 B 表示。当 B 接近人的两只眼睛的距离(2.5～3 英寸)时,这种配置模拟人的双目成像系统。如果我们把世界坐标的原点设在两台摄像机之间的中点,并假设两台摄像机有相同的焦距,用 F 表示,那么公式(12.2.1)与(12.2.2)中的关系将变成:

$$X_l=X+\frac{B}{2},X_r=X-\frac{B}{2},Y_l=Y_r=Y,Z_l=Z_r=Z \tag{12.2.6}$$

$$x_l=F\frac{X+B/2}{Z},x_r=F\frac{X-B/2}{Z},y_l=y_r=y=F\frac{Y}{Z} \tag{12.2.7}$$

我们也可以直接根据图 12.3(b)导出上述关系。位差矢量在这种情况下仅有水平分量,与深度的关系为:

$$d_x=x_l-x_r=\frac{FB}{Z} \tag{12.2.8}$$

可以由这些图像坐标按照下式恢复三维坐标:

$$X=\frac{B(x_l+x_r)}{2d_x},Y=\frac{By}{d_x},Z=\frac{FB}{d_x} \tag{12.2.9}$$

这些关系形成由位差信息导出深度从而导出三维结构信息的基础。

公式(12.2.8)在立体序列分析中起着重要的作用。可以从这个关系中导出几个很有趣的性质。首先,三维点(X,Y,Z)的位差值是与 X 和 Y 坐标不相关的,且与 Z 值成反比。物体点离摄像机越近,位差值就越大。对于远离摄像机的物体点,位差值减小。其次,位差值的范围随基线 B(即两台摄像机之间的距离)的增大而增加。最后,根据我们的位差定义,d_x 总是正的。这等价于对于同一个三维点,左坐标总是大于右坐标,从图 12.3 清楚可见。

图 12.4 示出了一个用平行摄像机获得的立体对[3]。很明显,左、右图像中的对应点在相同的水平线上,且较近的物体(较大的球)具有较大的水平位差。

12.2.3　会聚摄像机配置

在会聚立体配置中,两台摄像机的图像平面是互相倾斜的,使它们的焦线会聚于具有相同距离的一点,如图 12.5 所示。两台摄像机之间的角度称为会聚角。在图中,我们假设世界坐标原点在左、右坐标之间的中点,而 θ 是会聚角的一半。对于这种设置,左、右摄像机坐标与世界坐标的关系为:

---

① 严格地说,我们应该称它为图像位差,以便区别于视网膜位差和取景位差(指显示屏幕上的位差)。

② 注意,在前面几章中,我们已经用 $\mathbf{d}$ 来表示运动矢量。

③ 事实上,这里的两幅图像和图 12.6 的两幅图像是用射线跟踪技术人工生成的。

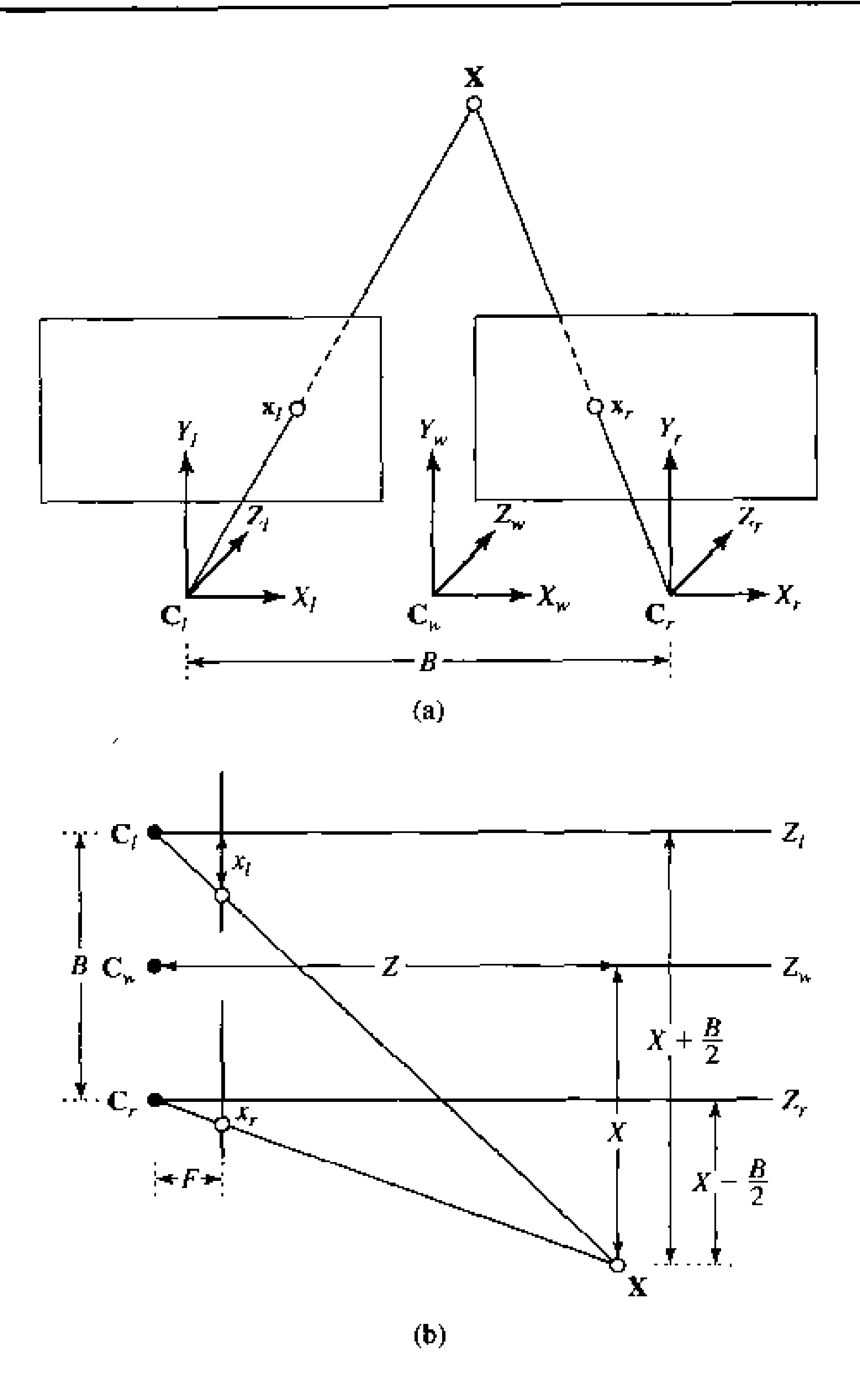


图 12.3 平行摄像机配置:(a)三维视图;(b) X-Z 横截面视图($Y=0$)

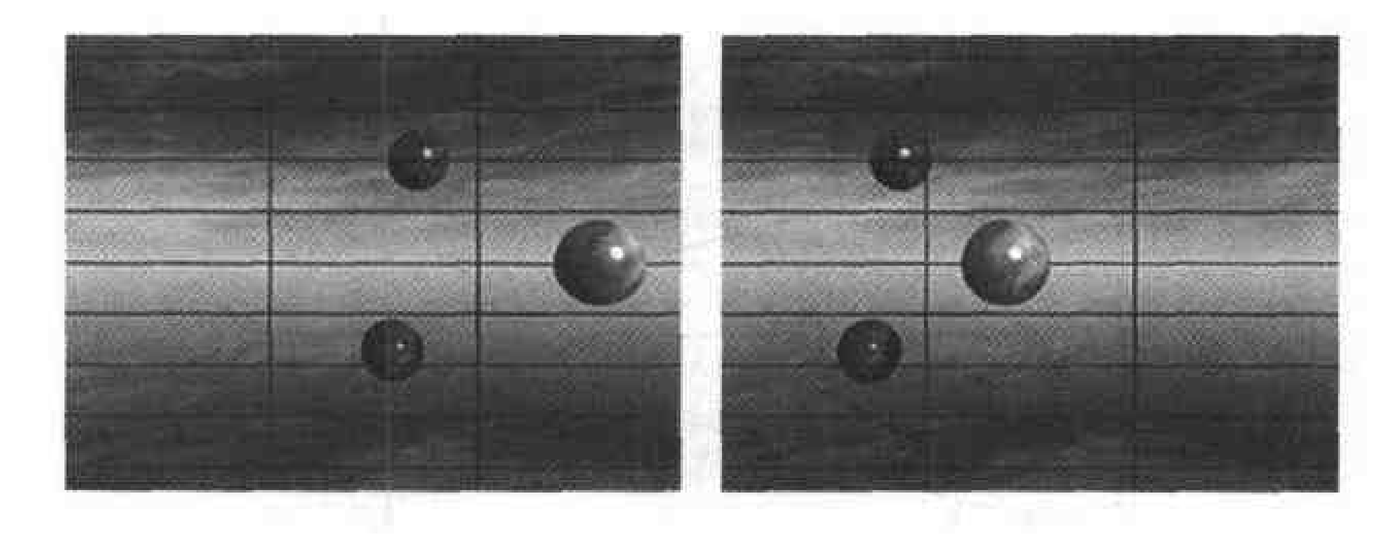


图 12.4 由两台平行摄像机采集到的立体图像对。如果分别用左眼和右眼看左图像和右图像,应该看到三个位于不同深度的球

$$\mathbf{R}_l = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}, \mathbf{T}_l = \begin{bmatrix} \cos\theta B/2 \\ 0 \\ \sin\theta B/2 \end{bmatrix} \tag{12.2.10}$$

$$\mathbf{R}_r = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \mathbf{T}_r = \begin{bmatrix} -\cos\theta B/2 \\ 0 \\ \sin\theta B/2 \end{bmatrix} \tag{12.2.11}$$

把上述关系代入公式(12.2.2)和(12.2.4)得到：

$$x_l = F\frac{\cos\theta(X + B/2) - \sin\theta Z}{\sin\theta(X + B/2) + \cos\theta Z}, \quad y_l = F\frac{Y}{\sin\theta(X + B/2) + \cos\theta Z}$$

$$x_r = F\frac{\cos\theta(X - B/2) + \sin\theta Z}{-\sin\theta(X - B/2) + \cos\theta Z}, \quad y_r = F\frac{Y}{-\sin\theta(X - B/2) + \cos\theta Z} \tag{12.2.12}$$

我们也可以根据图 12.5(b)所示导出上述关系。由这些公式,可以导出水平(和垂直)位差与三维世界坐标(X, Y, Z)之间的关系。这个关系不如平行摄像机情况下的直接。当 θ 较小(小于 1°)时,可忽略垂直位差。

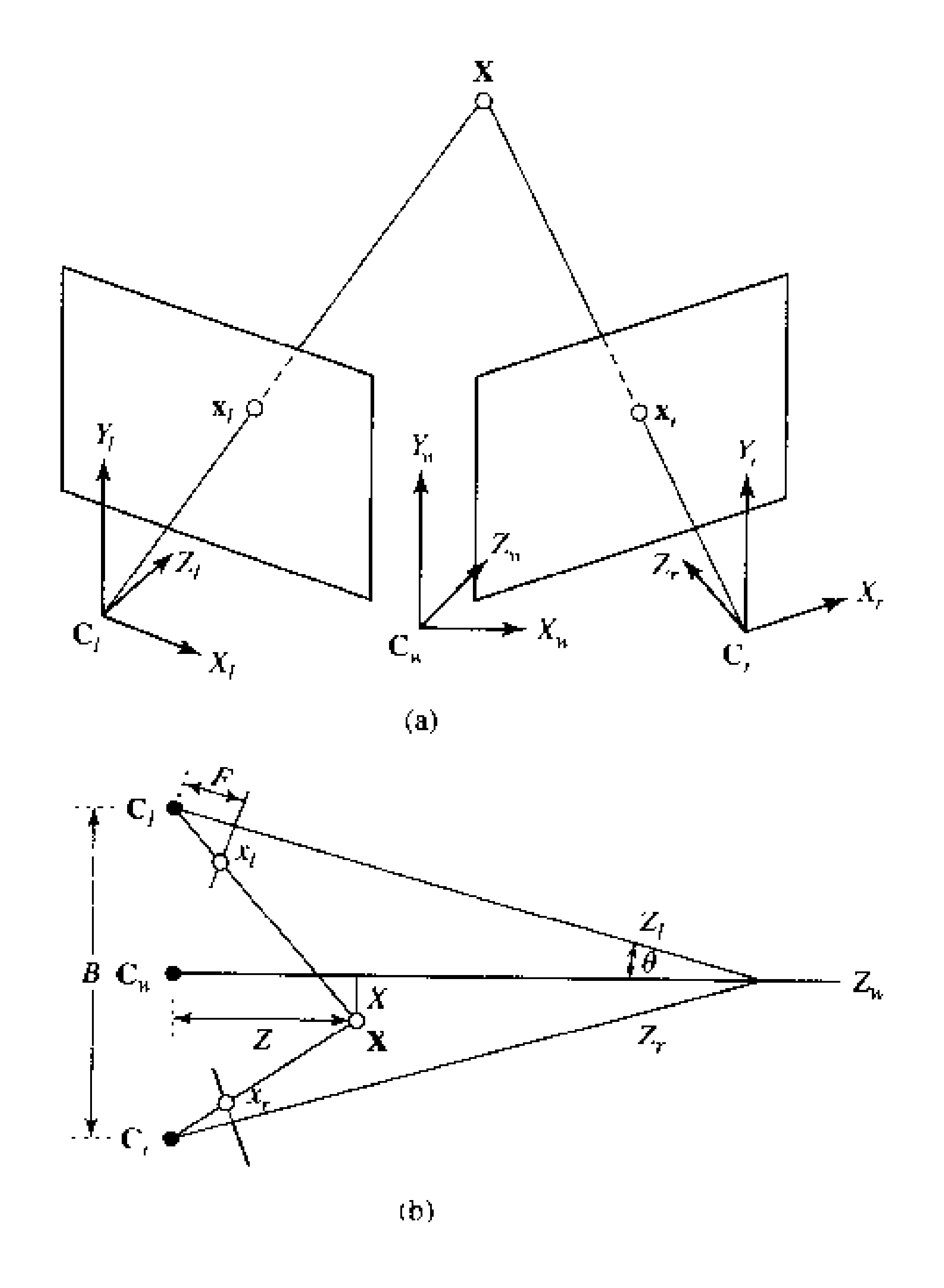


图 12.5 会聚摄像机配置:(a) 三维视图;(b) X-Z 横截面视图($Y=0$)

应用会聚配置,成像系统对接近摄像机的物体可得到比平行配置更好的深度感觉。然而,

当由会聚立体摄像机采集的立体对被直接投影到屏幕上观看时,感觉到的深度是失真的。一个众所周知的效应是梯形失真。这指的是这样一个事实:当摄取一个具有相等间隔线的矩形点阵时,左摄像机显现的两条相邻线之间的垂直距离好像左边的比右边的大(右摄像机所生成的图像有相反效果)。一个由会聚摄像机获得的立体对的例子示于图 12.6。当用平行投影配置显示并用人眼观看时,这样的垂直位差将引起深度感觉的问题。为了显示会聚摄像机系统采集的图像,需要几何校正(也称为图像校正),以便使校正的图像看起来好像是用平行设置采集的。

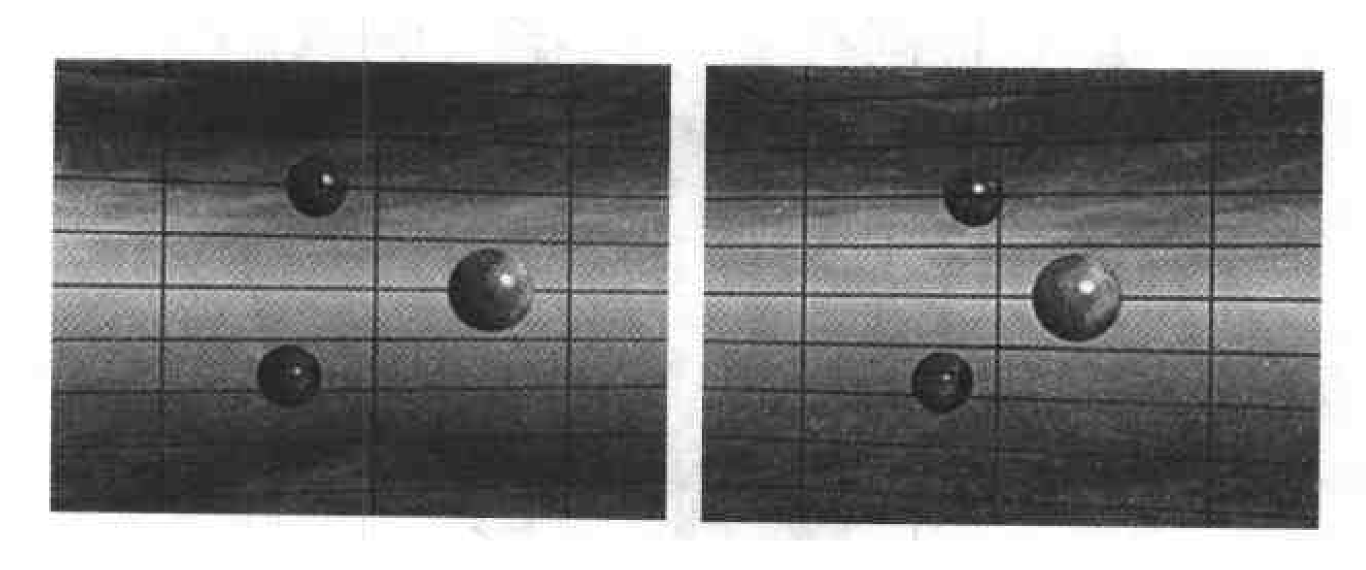


图 12.6 一个用会聚摄像机获得的立体对的例子(注意梯形效应)

12.2.4 外极几何

图 12.7 示出了任意立体摄像机配置中三维点的左右图像之间的关系。用 $\mathbf{X}$ 表示任意选择的三维点,由该点以及左右摄像机的光学中心 $\mathbf{C}_l$ 和 $\mathbf{C}_r$ 定义的平面 Π 称为外极平面。这个平面与左右图像平面 Π_l 和 Π_r 的交线称为左(连接 $\mathbf{x}_l$ 和 $\mathbf{e}_l$ 的线)右(连接 $\mathbf{x}_r$ 和 $\mathbf{e}_r$ 的线)外极线,分别用 ep_l 和 ep_r 表示。$\mathbf{C}_l$ 在右图像中的像 $\mathbf{e}_r$ 称为右外极。类似地,$\mathbf{C}_r$ 在左图像中的像 $\mathbf{e}_l$ 称为左外极。我们看到,对于落在左外极线上的任何成像点,它在右图像中的对应像素一定在右外极线上。这就是所谓的外极约束。可用这个性质来约束位差估计的搜索范围,如在 12.3 节中所讨论的。

注意,存在一个外极平面和两条相应的与每个所选择的图像点有关的外极线。例如,给定左图像上的一点 $\mathbf{x}_l$,它的左外极线是连接该点与左外极 $\mathbf{e}_l$ 的线。它相应的右外极线是连接该点在右视图上的像 $\mathbf{x}_r$ 与右外极($\mathbf{e}_r$)的线。所有左外极线都通过左外极,而所有右外极线都通过右外极。

图像点与它的外极线之间的关系取决于摄像机设置。立体成像中的一个非常重要的结果是这种关系可以用一个称为基本矩阵的 3×3 矩阵$[\mathbf{F}]$完全地表征。令 $\tilde{\mathbf{x}}_i^T=[\mathbf{x}_i^T,1]$, $i=l,r$,其中,$\mathbf{x}_l$ 和 $\mathbf{x}_r$ 表示同一个三维点的左像和右像[①]。它们的关系为:

$$\tilde{\mathbf{x}}_r^T[\mathbf{F}]\tilde{\mathbf{x}}_l=0,\quad \tilde{\mathbf{x}}_l^T[\mathbf{F}]^T\tilde{\mathbf{x}}_r=0 \tag{12.2.13}$$

① $\tilde{\mathbf{x}}^T=[\mathbf{x}^T,1]$是 $\mathbf{x}$ 在齐次坐标(也称为投影坐标)中的表示。用齐次坐标可把大多数在透视成像中所遇到的非线性关系转换成线性关系。

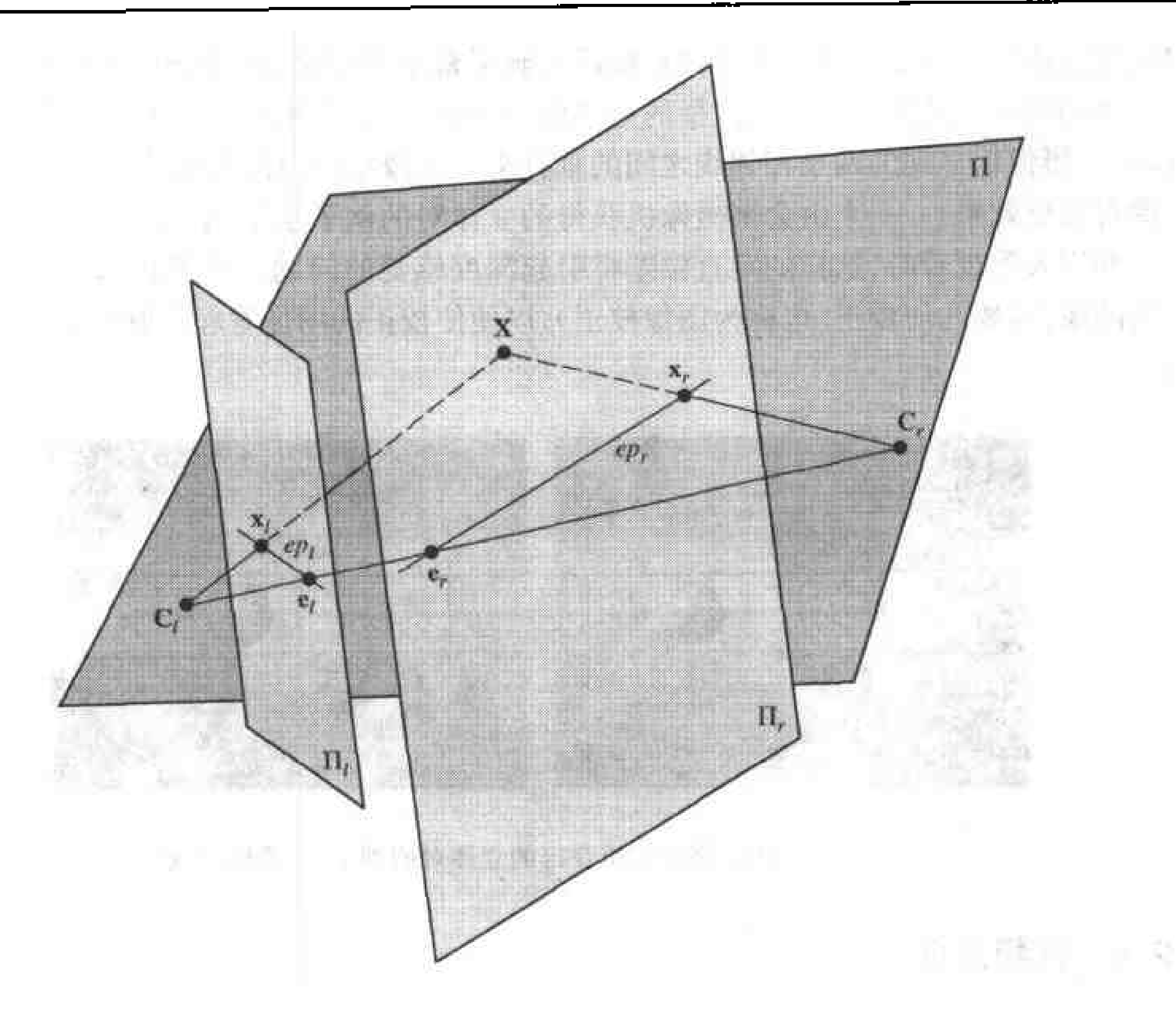


图 12.7 任意摄像机配置的外极几何学

在图 12.2 所示的简化配置中,基本矩阵仅取决于两个摄像机坐标的相对关系,$[\mathbf{R}]_{rl}$,$\mathbf{t}_{rl}$①。一般地,它也将取决于摄像机的固有参数。关于上述结果以及[F]与摄像机参数之间的关系的推导,读者可参考 Faugeras 的优秀著作[3]。当摄像机参数未知时,[F]中的元素可以基于公式(12.2.13),由一组在左、右图像中检测到的对应点来估计。关于这一课题可参阅参考文献[23]。

对于平行摄像机设置,外极在无穷远点而外极线都是水平的,如图 12.8 所示。对于右图像中任意给定的点 $\mathbf{x}_r$,与该点有关的左右外极线简单地是与该点有相同 y 坐标的水平线。另一方面,对于会聚配置,左右外极线是互相倾斜的。这可以很容易地在图 12.6 中所给出的示例图像中观察到。然而,当角 θ 足够小时,外极线可以作为水平的看待。

给定由两台会聚摄像机所获得的一对图像,可校正它们,使校正后的图像看起来像是用两台平行摄像机采集的。在视觉上,其效果是对图像对的映射,使得原来倾斜的一组外极线变成水平的和平行的。这种校正可大大简化位差估计问题。对于用平行投影系统显示这类图像也需要这种校正。从上述讨论回想一下,外极约束是由联系两台摄像机的基本矩阵完全表征的。因此,校正过程的关键是确定基本矩阵。实际上,可能不会精确地知道摄像机配置参数,从而必须由两个给定图像中对应的特征点来确定基本矩阵。讨论摄像机校准和校正的技术超出了本章的范围;读者可参阅参考文献[19,3]以较好地了解这一课题。

---

① 基本矩阵把左、右图像中对应的点通过公式(12.2.13)联系起来。在第 7 章,我们了解到旋转和位移前后的三维物体图像是由本质矩阵通过公式(7.1.11)联系的。显然,如果我们把$[\mathbf{R}]_{rl}$和 $\mathbf{T}_{rl}$ 作为成像物体的运动参数,则基本矩阵等价于本质矩阵。

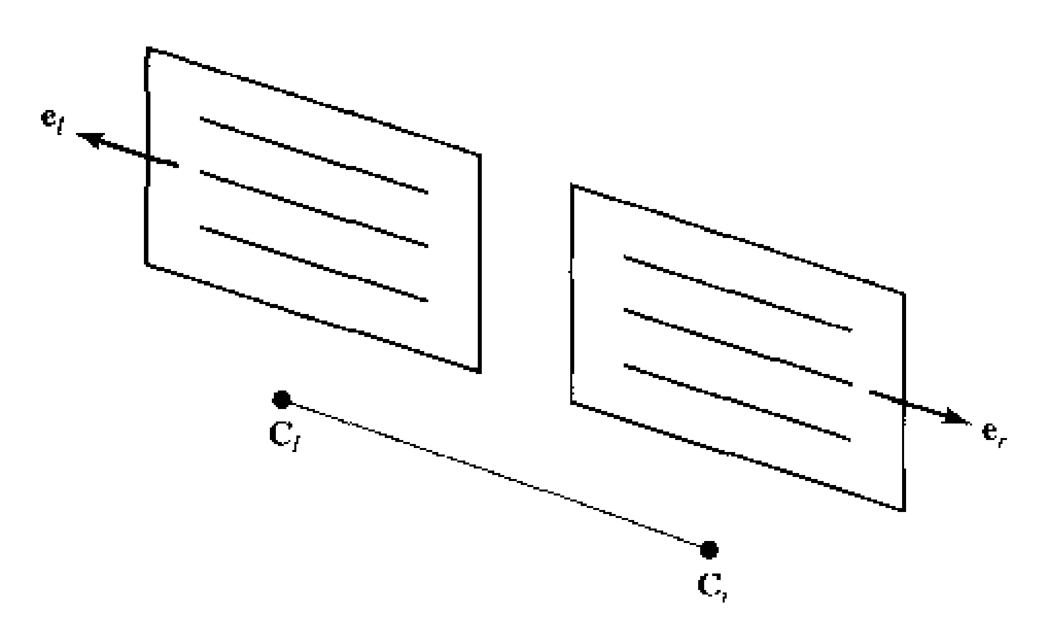


图 12.8　平行摄像机配置的外极几何：外极在无穷远点，而外极线是平行的

12.3　位差估计

如前面一节所介绍的，位差是指成像场景中一个三维点在左右图像平面中的二维坐标之间的位移。如在那里所说明的，根据与立体图像对中两个对应点有关的位差，可导出与该三维点有关的深度，从而导出它在真实世界中完全的三维坐标。由于这个原因，位差估计在任何三维图像或视频处理系统中都是重要的一步。它对于在给定视图之间生成中间视图也是重要的。

位差估计问题与运动估计问题类似，它们都需要在两幅图像的像素之间建立对应关系。在非基于特征的方法中，对于一幅图像中的每个点，任务是要确定它在另一幅图像中的对应点，或当它在另一幅图像中被遮挡时标示它。对于基于特征的方法，任务是首先在一幅图像中检测某些特征点，然后寻找它们在另一幅图像中的对应位置。在运动估计中，两幅图像在时间上是不同的，而在位差估计中，两幅图像是同时但从不同的摄像机视点取得的。

位差估计比运动估计更复杂且更具有挑战性。第一，一般在场景中仅有有限数目的物体运动，因此对大多数像素 MV 都是零。然而，在立体成像中，除了那些远离图像平面的像素以外，几乎所有的像素都有非零位差。第二，MV 的范围一般是相当有限的，而对于接近摄像机的物体，位差矢量(DV)可能非常大。例如，对于 BT.601 视频，MV 一般都不会超过几个像素，而 DV 在水平方向上可达到 40～50 个像素，而在垂直方向上达到 5 个像素(假设采用稍微会聚的摄像机配置)[6]。显然，如果用类似于运动估计的块匹配算法的搜索技术，搜索区域必须大大增加。块效应也会更明显，因为块状的恒定模型不能很好地近似位差变化。

除了在位差场中存在更多的如下面所描述的物理上引入的约束外，位差估计的一般方法非常类似于运动估计。基本上，必须建立一个目标函数来测量一个给定的视图与它用位差补偿预测(DCP)由其他视图预测的版本之间的误差。如果我们假设位差场是平滑的，那么可把一些平滑性测度加到目标函数中去。在物体边界或遮挡区域这些平滑性约束应该是宽松的。估计的参数进一步受到一些物理约束。各种方法之间的不同之处在于位差场的参数化以及搜索方法。在下面，我们首先描述一些对位差估计有用的重要的约束，然后给出几种估计方法。

12.3.1　位差分布的约束

DV 是由摄像机几何和物体表面的连续性约束的。这些约束如果明智地使用，则在位差估计中是非常有益的。

外极约束 对于一个给定的立体成像配置,外极约束是指立体对中对应的像素总是在各自的外极线上。回想前面讲过的内容,如果已知基本矩阵,那么给定(比如右图像中的)任意一点,可用公式(12.2.13)确定左图像中的外极线,这个公式定义了该点在左图像中所有可能的位置。因此,我们只需沿线搜索。

对于平行摄像机配置,外极线与图像的水平扫描线平行,从而我们可以把搜索限制在右图像点所在的水平线上。当摄像机配置是非平行时,尽管可以把搜索限制在相应的实际摄像机配置的外极线上,但一个更简单的方法是先把左右图像映射成平行的配置,然后应用平行配置的步骤[21]。在每一种情况下,我们必须先估计摄像机的几何参数或基本矩阵,它对于确定外极线以及对于再投影是必需的。

平行摄像机的单向性 如 12.2.2 节所述,对于平行摄像机配置,DV 只有水平分量且总是为正(即左水平坐标总是大于右水平坐标)。

顺序约束 假设成像物体是不透明的,则物体点的相对位置在场景的两个视图中是相同的:若左视图中一个特征点在另一个特征点的左边,则在右视图中它也在左边。设 $x_{r,1}$ 和 $x_{r,2}$ 表示在右图像中同一个水平线上的两点,且假设 $x_{r,1} < x_{r,2}$。则通过要求 $x_{l,1} < x_{l,2}$,我们有 $d_{x,2} > d_{x,1} + x_{r,1} - x_{r,2}$,它在给定前面估计的 $d_{x,1}$ 条件下限制了 $d_{x,2}$ 的搜索范围。特别地,对于右图像中的相邻像素(即,$x_{r,2} = x_{r,1} + 1$),它们的位差关系为:$d_{x,2} > d_{x,1} - 1$。

12.3.2 位差函数模型

对于给定的摄影配置,位差是深度的确定性函数。给定一个实际表面函数的模型,可导出位差函数对应的模型。这样的模型有助于位差估计。

考虑最简单的情况,其中,成像场景的表面可由一个平面近似,从而有:

$$Z(X,Y) = aX + bY + c \tag{12.3.1}$$

用公式(12.2.6)和(12.2.7),我们可用右图像坐标重写 Z:

$$Z(x_r,y_r) = a\left(\frac{Zx_r}{F} + \frac{B}{2}\right) + b\frac{Zy_r}{F} + c \tag{12.3.2}$$

或

$$Z(x_r,y_r) = \frac{c + aB/2}{1 - ax_r/F - by_r/F} \tag{12.3.3}$$

可以看到,表面函数随图像坐标成非线性变化。把上述结果代入公式(12.2.8)得:

$$d_x(x_r,y_r) = \frac{1}{c/B + a/2}(F - ax_r - by_r) \tag{12.3.4}$$

因此,当表面是一平面时,位差函数在图像坐标中是仿射的。

事实上,成像场景的整个表面(即深度分布)不能由一个平面很好地近似。然而,它一般可分成很小的小块,使得每个小块是近似平坦的。用上述结果,我们可以把每个小块上的位差函数模型化为仿射。这样位差估计问题就转变成对每个小块的三个仿射参数的估计,等价于在每个小块的三个角点(节点)处估计位差(只估计 d_x)。如果我们用三角形网格表示整个表面,那么位差估计问题可简化为估计节点处的位差。然后,每个小块中的位差函数通过使用仿射模型从节点中内插来获得。

用类似的方法,我们也可导出较高阶(弯曲的)表面的位差模型。这样的参数模型对位差

估计非常有用。

12.3.3 基于块的方法

这种方法假设参考视图的每个小方块上的位差函数可描述为一个常量或一个低阶多项式。基于估计的位差函数(即位差补偿),通过使扭曲后的两视图之间的误差最小来确定常量位差或多项式系数。可采用在基于块的运动估计中使用的穷尽或梯形下降搜索法。但搜索应该是适当受限的,使用12.3.1节中所描述的约束。在假设位差为常数的特殊情况下,只须搜索单个水平位移,且只需在一个方向上搜索,沿正方向还是负方向取决于用哪个视图作为基准。然而,与运动估计相比较,如果成像场景中的深度变化相对较大,或摄像机之间的基线分离较大,则搜索范围应该大得多。为了克服与潜在的非常大的位差值有关的困难,应该采用例如6.9节中所描述的分层搜索策略。基于块的估计方法的一个好的综述可以在参考文献[20]中找到。

与运动估计不同,在运动估计中块状恒定模型对于大量的块是相当精确的,而这种模型只适合相应块的表面小块是与成像平面平行的平坦表面的情况,但这种情况很少。然而,当块尺寸足够小时,仿射模型是相当好的。在这种情况下,相应于每个块的表面小块近似平坦,并且(如12.3.2节所示)仿射函数很好地描述平坦小块上的位差函数。

12.3.4 二维网格的方法

除块状恒定模型外,在运动估计中所描述的基于网格的方法也可用于位差估计。在这种情况下,首先在基准视图(比如左视图)中应用网格,并试图在右视图中找到对应的节点位置。注意,每对相应的二维网格元素可认为是一个三维表面小块在左右图像上的投影,如图12.9所示。与运动估计一样,最好用左右视图之间的节点位移来参数化位差函数,而不用仿射或双线性模型中的多项式系数。类似于基于网格的运动估计方法(6.6节),可通过使对应元素之间的位差补偿预测误差最小来估计节点位差。如果图像是用平行配置获得的,只需要搜索水平位差。

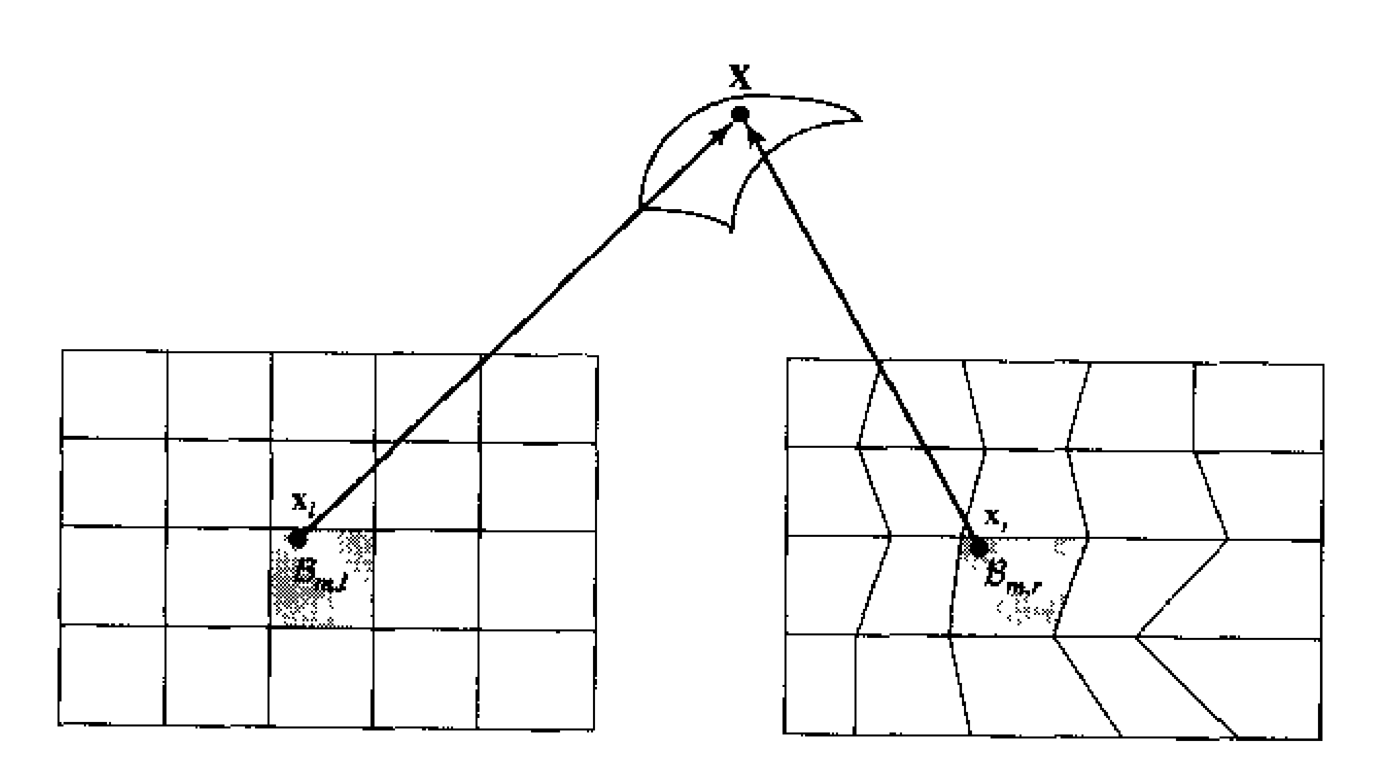


图12.9 三维网格与二维网格之间的对应

在参考文献[21]中描述了一种这样的方法。如图12.9所示,用一个矩形网格描述左视图,根据左视图中对应的节点水平地移动右视图中的节点。每个元素内的位差函数被模型化为双线性的,

从而对应于这个元素的三维小块是弯曲的表面。为了确定每个节点的水平位差，与该节点相连的四个元素求和获得的位差补偿误差被最小化。不进行穷尽搜索，而使用基于梯度的快速算法。

图 12.10 比较了对于一个立体图像对样本（“人”），用基于块的方法和上述基于网格的方法[21]进行位差估计的结果。这个图像对是用基线距离为 50 cm 的两台平行摄像机获得的。这种大的基线距离导致两个图像之间有非常大的位差值。先估计全局位差，并在局部位差估计之前用全局位差补偿右图像。块匹配算法（假设每个块内是恒定位差）用 16×16 个块和 ±100个像素的搜索范围。基于网格的方法使用一个元素尺寸为 32×16 的左图像上的矩形网格，并在右图像中水平地移动节点位置，以使对应元素之间的位差补偿误差为最小。尽管块匹配算法对于预测图像产生较高的 PSNR，但基于网格的方法将产生视觉上更精确的预测。

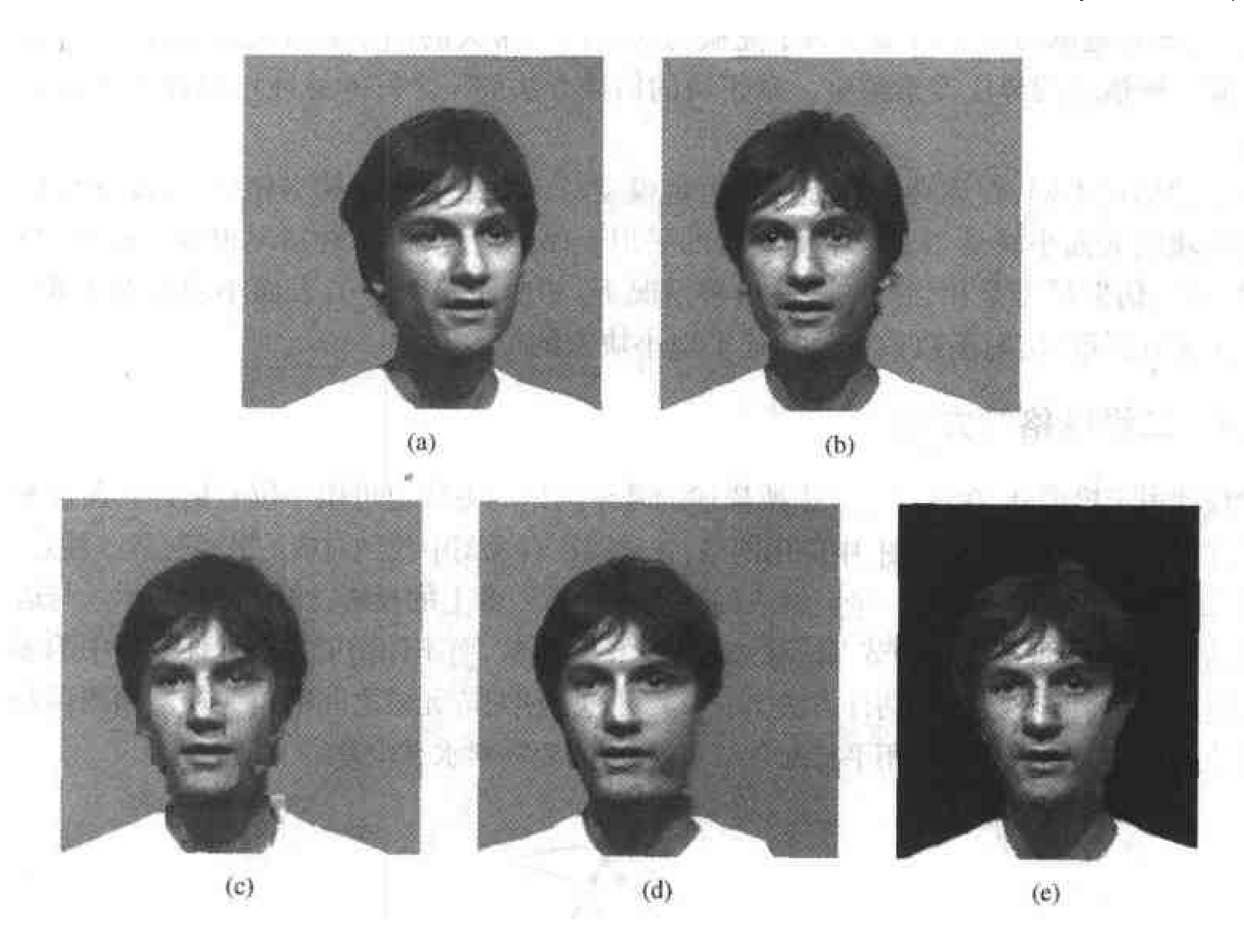


图 12.10 (a)原始左图像；(b)原始右图像；(c)由块匹配算法获得的预测右图像（前景上的PSNR为32.03dB）；(d)由基于网格的算法获得的预测右图像（前景上的PSNR为27.28dB）；(e)用参考文献[2]的动态编程方法获得的预测右图像

这个方法在整个基准图像上覆盖规则的网格。这将在物体边界附近和深度不连续的地方引起估计误差。一个较好的方法（尽管较复杂）是生成一个跟随物体轮廓的网格，且允许不连续的物体边界。例如，如果生成的网格使每个网格元素对应于一个近似平坦的表面小块，那么这个元素上的位差函数可用仿射函数精确地模型化，如 12.3.2 节所示。在这种情况下，三角形网格将是较适合的。

采用基于网格的位差估计的好处是容易用产生的位差信息构造中间视图。这个主题将在 12.4 节中进一步讨论。

12.3.5 利用动态编程的线内边缘匹配

根据平行成像几何(或在图像校正后),外极线是水平扫描线。因此,给定一条水平扫描线,应该只在这条线内搜索左右图像中的相应的点对。不是在所有像素之间搜索匹配,我们可以寻找这条线上的所有边缘点,并试图把右图像上的每个边缘像素与左图像上的一个边缘像素进行匹配。沿着这条线搜索匹配对可以转化为在二维搜索平面上寻找一条路径的问题,该平面的垂直和水平轴是右和左边缘位置,如图 12.11 所示。该图假设在左、右图像中分别有 M 和 N 个边缘点。每个节点对应右边缘点与左边缘点的结合。一般地,不是所有在右图像中看到的边缘像素也都将出现在左图像中。并且,右图像中的一个边缘像素可能对应左图像中的几个边缘点或一个点也不对应。这些情况发生在物体的某些部分在一个视图中可观察到而在另一个视图中被遮挡的时候。因此,匹配可能是一对多、多对一或一对零的。然而,合理的匹配应该满足顺序约束。从右图像中的最左边的边缘像素开始,对于右图像中的每个新的边缘像素,它在左图像中的匹配边缘点必须在剩余的未匹配边缘当中或是在前一步匹配的那一个。因此,合理的路径是不返回的,无论是在水平方向还是在垂直方向上。如果我们在图的每个可能的节点分配一个代价,则该问题就是寻找从起始对(m_0)到终止对(m_e)的最佳路径,使总的代价为最小。例如,对于给定的节点,代价可确定为像素之间的平方或绝对误差之和,这些像素是在定义该节点的两个边缘像素周围的一个小窗中的对应像素。它也可能包括在当前节点和前一个节点之间的对应像素之间的误差。

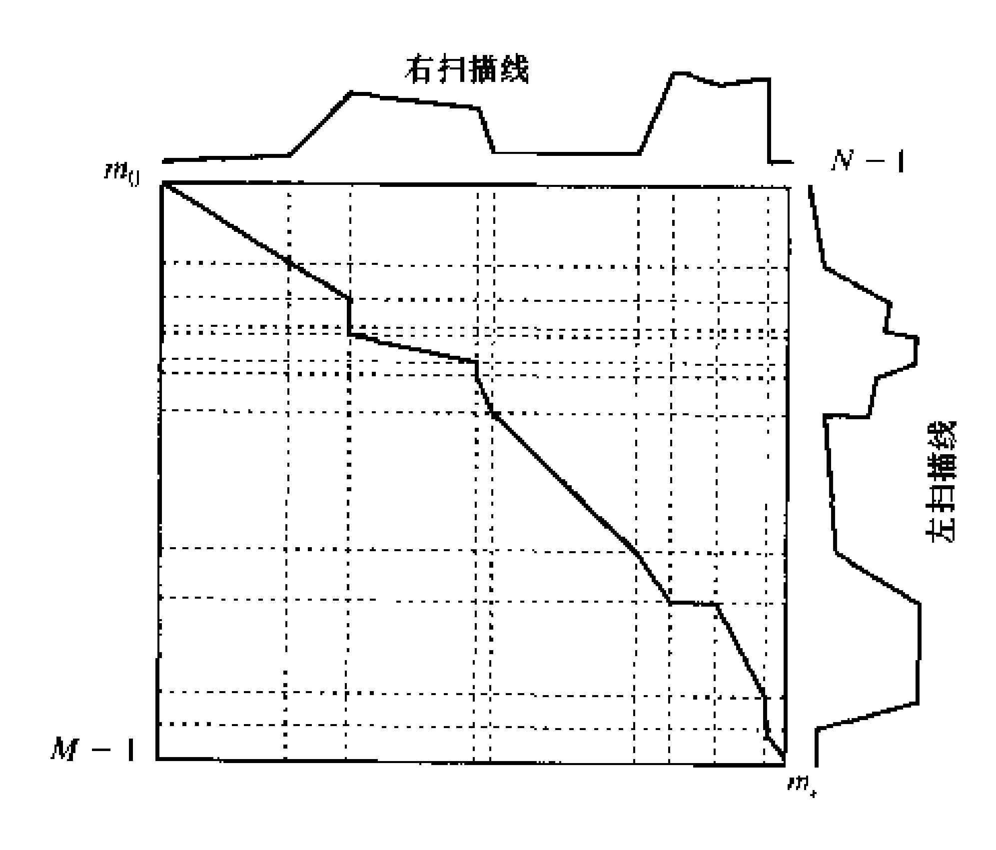


图 12.11　立体匹配问题中的路径寻找模拟

上述方法没有考虑扫描线之间边缘像素的相互依赖性。如果有一个垂直边缘延伸跨过扫描线,那么在一条扫描线中的对应与相邻扫描线中的对应具有很强的相关性。特别地,如果在右图像中两个点位于垂直连接的边缘上,则很可能在左图像中它们对应的点也位于垂直连接的边缘上。促进扫描线之间的这种一致性的一种方法是在代价定义中包括一个惩罚项,对那些与在前面的线中已发现的匹配不一致的匹配赋以更高的代价。通过这种修正,可把立体匹配问题模型化为寻找一个最佳地满足扫描线内匹配和扫描线间一致性的表面。这里,匹配表面是由二维匹配路径的堆积定义的。

Ohta 和 Kanade[10]首先提出把位差估计问题模型化为一个表面匹配问题，如这里所概括的。而且，他们用动态编程方法开发了一种快速搜索算法。Ohta 和 Kanade 的原始工作只考虑了由沿垂直方向的边缘所施加的约束。Falkenhagen 把这个方案扩展到考虑更多的二维约束[2]。图 12.10(e)示出了用这个方法对"人"所预测的右图像。

12.3.6 结构和运动的联合估计

上述讨论集中于由同时获得的一对立体视图来估计位差函数。由位差函数，我们可导出成像场景的结构(即空间分布)，如 12.2 节中所描述的。它允许我们重建成像物体的三维表面。对于立体序列，我们不仅对给定帧的结构估计感兴趣，也对帧间的三维运动估计感兴趣。最具有挑战性的问题是联合地估计结构和运动。

针对这个问题的一个通常的方法是用三维网格(也称为线框)模仿成像物体的表面。这样，结构和运动估计问题是要寻找在初始帧中所有节点的三维坐标和相继帧间的三维节点的位移。注意，三维网格将在左右图像中投影成二维网格，如图 12.12 所示。为了确定初始的三维网格，我们可先在一个视图中构造二维网格，然后再在另一个视图中寻找对应节点的位置。这可用 12.3.4 节中所描述的基于网格的位差估计方法实现。由对应节点之间的位差，我们可确定所有节点的三维位置。为了使生成的三维网格符合实际物体的表面，必须使初始的二维网格匹配于物体的边界，且每个元素应该对应于一个平坦的小块。然而，这要求知道物体结构。因此，重要的是基于立体视图获得好的物体分割。同时，可能需要一个迭代过程，每次迭代基于前一个物体结构的估计，从一个修改的新的二维网格开始。为了确定三维节点的位移，我们可以确定两个视图中的二维节点的位移，通过使同一个视图的两个相邻帧之间的运动补偿预测误差为最小，以及使相同帧时的两个视图之间的位差补偿预测误差为最小来实现。结构和运动估计中的主要困难在于如何处理遮挡的区域，包括那些在同一个帧时只在一个视图中看到的区域，和随着时间消失或新出现的区域。关于这个课题的研究论文，见参考文献[5,7,18]。

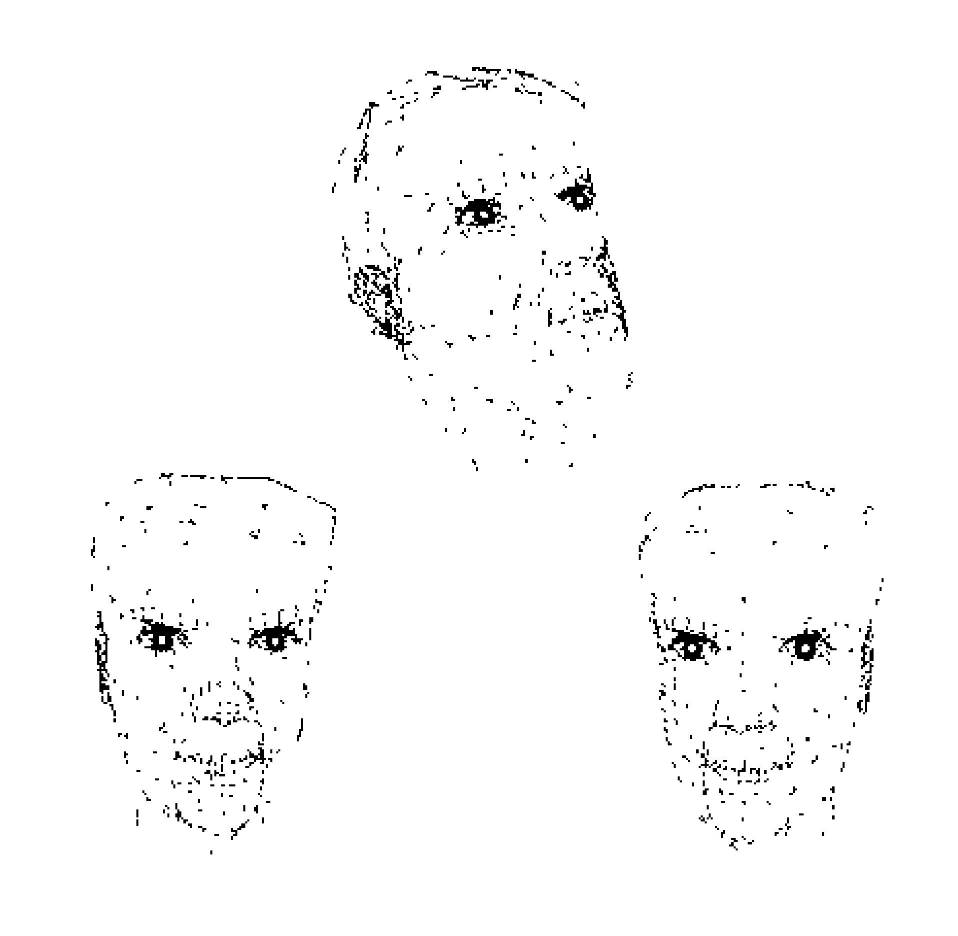


图 12.12 三维表面网格投影到左右图像上的二维网格

12.4 中间视图合成

一个有趣的立体处理任务是由给定的视图来内插或合成中间视图——例如，由左右视图生成中央视图。这在虚拟现实显示中是经常需要的，这时所显示的视图必须根据用户的视点经常地更新。在先进的立体或多视视频编码系统中也需要视图合成，这时首先由参考视图合成欲编码的视图，然后对合成残差图像进行编码。一个不考虑位差的自然视图合成方法是线性内插，由左、右视图$\psi_l(\mathbf{x})$和$\psi_r(\mathbf{x})$，通过下式生成内插的中央视图$\psi_c(\mathbf{x})$：

$$\psi_c(\mathbf{x}) = w_l(\mathbf{x})\psi_l(\mathbf{x}) + w_r(\mathbf{x})\psi_r(\mathbf{x}) \tag{12.4.1}$$

如果由中央到左视图的基线距离是 D_{cl}，而到右视图的基线距离是 D_{cr}，则按照下式确定加权因子：

$$w_l(\mathbf{x}) = \frac{D_{cr}}{D_{cl} + D_{cr}},\ w_r(\mathbf{x}) = 1 - w_l(\mathbf{x}) \tag{12.4.2}$$

这种方法尽管简单，但将不能得到满意的结果，因为在不同视图中具有相同图像坐标的像素对应不同的物体点，它们之间的平均将产生模糊的图像。

图 12.13 给出了更适当的位差补偿内插。设 $\mathbf{d}_{cl}(\mathbf{x})$和 $\mathbf{d}_{cr}(\mathbf{x})$分别表示由中央到左视图和到右视图的位差场。内插的中央视图按照下式确定：

$$\psi_c(\mathbf{x}) = w_l(\mathbf{x})\psi_l(\mathbf{x} + \mathbf{d}_{cl}(\mathbf{x})) + w_r(\mathbf{x})\psi_r(\mathbf{x} + \mathbf{d}_{cr}(\mathbf{x})) \tag{12.4.3}$$

对于在两个视图中都可见的像素，加权因子可按公式(12.4.2)选择。如果像素只在一个视图中可见，那么另一个视图的加权因子应该为零。具体地，

$$w_l(\mathbf{x}) = \begin{cases} \dfrac{D_{cr}}{D_{cl} + D_{cr}}, & \text{当 } \mathbf{x} \text{ 在两个视图都可见} \\ 1, & \text{当 } \mathbf{x} \text{ 只在左视图可见} \\ 0, & \text{当 } \mathbf{x} \text{ 只在右视图可见} \end{cases} \tag{12.4.4}$$

例如，如果成像物体是一张脸，那么对于脸的中央部分，两个视图都将用于内插。对于脸的左(右)部分，应该只使用左(右)视图。

以上描述的方法假设 $\mathbf{d}_{cr}(\mathbf{x})$和 $\mathbf{d}_{cl}(\mathbf{x})$是已知的。实际上，根据给定的左图像和右图像，我们只能估计比如由左视图到右视图的位差 $\mathbf{d}_{lr}(\mathbf{x})$。要由 $\mathbf{d}_{lr}(\mathbf{x})$得到 $\mathbf{d}_{cl}(\mathbf{x})$和 $\mathbf{d}_{cr}(\mathbf{x})$不是一个简单的工作。注意，通过下式容易内插出由左视图到中央视图的位差场：

$$\mathbf{d}_{lc}(\mathbf{x}) = \frac{D_{cr}}{D_{cl} + D_{cr}}\mathbf{d}_{lr}(\mathbf{x}) \tag{12.4.5}$$

但这不允许我们为中央视图的每个像素在左视图中找到对应点。确切地说，它为左视图中的每个像素确定了中央视图中的对应点。一般地，会存在左视图中未包含的像素，或对应左视图中一个以上像素的像素。

如果 $\mathbf{d}_{lr}(\mathbf{x})$ 是用基于网格的方法估计的，则这个问题是可行的。在这种情况下，可以很容易地由左视图和右视图中的网格生成中间视图的网格，如图 12.14 所示。具体地说，中央视图中的节点位置 $\mathbf{x}_{c,n}$是由左视图和右视图中的节点位置 $\mathbf{x}_{l,n}$和 $\mathbf{x}_{r,n}$用下式生成的：

$$\mathbf{x}_{c,n} = \frac{D_{cr}}{D_{cl} + D_{cr}}\mathbf{x}_{l,n} + \frac{D_{cl}}{D_{cl} + D_{cr}}\mathbf{x}_{r,n} \tag{12.4.6}$$

这样，对于中央视图的一个元素的每个像素，通过使用元素的形状函数(见6.6节)的节点位置内插，可确定它在左(或右)视图中对应的点。在平行成像几何情况下，只存在水平位差，问题就更为简单。对这种情况的清楚描述见参考文献[21]。图12.15给出了用这种方法对立体对"人"的内插结果。

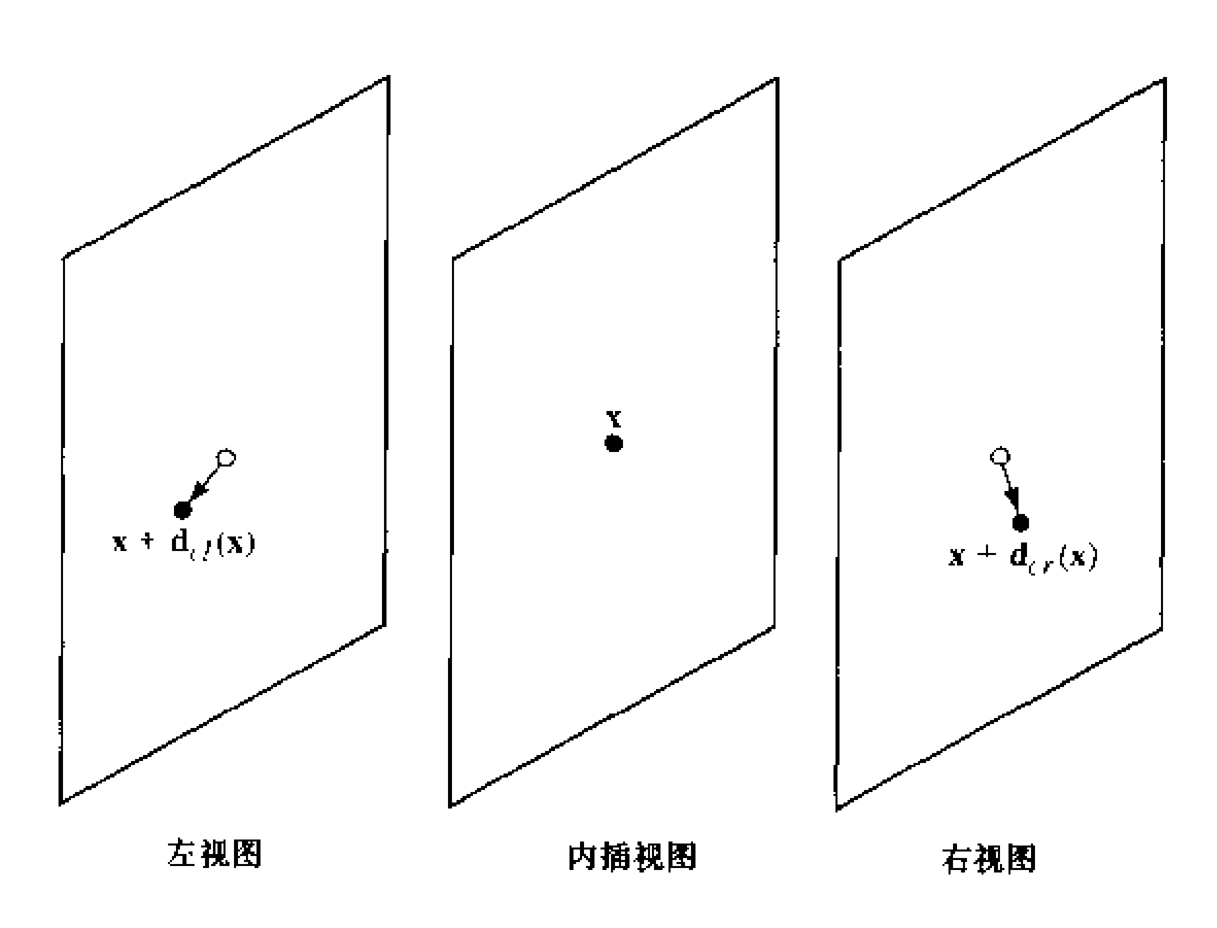


图12.13 位差补偿内插：由左视图中的 $\mathbf{x}+\mathbf{d}_{cl}(\mathbf{x})$ 和右视图中的 $\mathbf{x}+\mathbf{d}_{cr}(\mathbf{x})$ 内插 $\mathbf{x}$

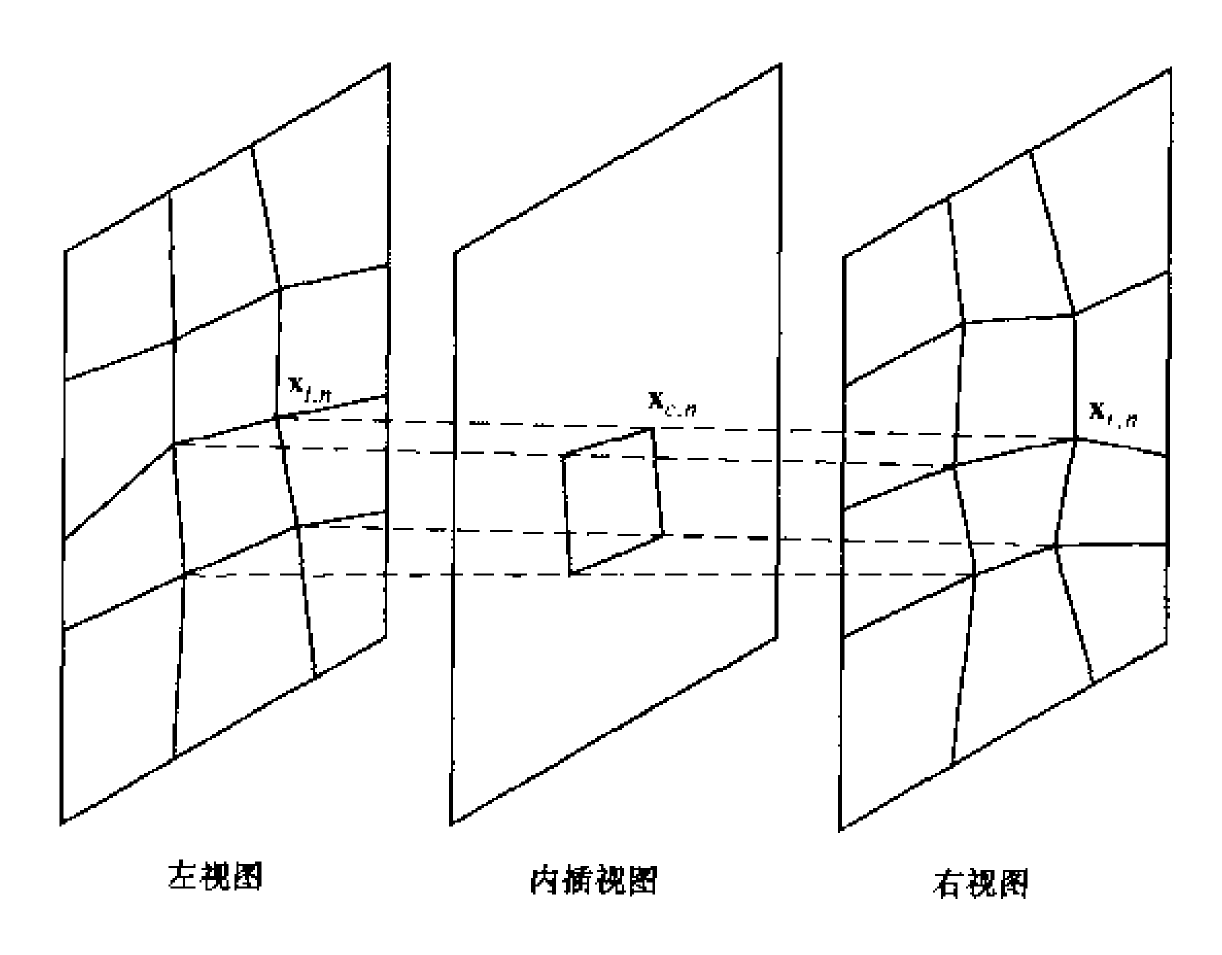


图12.14 通过对左、右视图中的节点位置进行线性内插来生成中央视图的网格





图 12.15　一个中间视图合成的例子。由左右图像来内插中央图像。该结果是通过用基于网格方法估计的左右视图之间的位差图，用基于网格的位差补偿内插获得

12.5　立体序列编码

立体和多视序列(包括三维 TV)应用中的一个主要困难是立体和多视序列的数据量非常大。为了以合理的代价存储或传输立体序列，必须实现数据的实质上的压缩。在本节中，我们给出为立体序列开发的几种压缩方案。

12.5.1　基于块的编码和 MPEG-2 的多视类

在这个方法中，编码器首先用单视视频编码算法压缩左视图序列。对于右视图序列，每一帧由左视图序列对应的帧基于估计的位差场进行预测，并且对位差场和预测残差图像进行编码。位差估计和补偿过程都采用基于块的方法；也就是说，对于右视图中的每个图像块，它们在左视图中寻找一个最相似的块。一个较好的方法是根据哪一个给出较小的预测误差，在相同时间的不同视图之间的位差补偿预测与相同视图的不同帧之间的运动补偿预测之间进行切换。

这种方法已经在 MPEG-2 标准的多视类中被采用[4]，它可用该标准的时间可分级性模式实现。如在 13.5.3 节中所讨论的(图 13.24)，用 MPEG-2 的时间可分级性，把序列分成两个亚采样帧的集。基本层(比如，由偶帧组成)用单向 MCP 进行编码，其中参考帧是来自前面编码的偶帧。由剩余的奇帧组成的增强层用双向 MCP 进行编码，一个参考帧来自前面编码的奇帧，而另一个来自基本层中一个靠近的偶帧。图 12.16 说明了时间可分级性如何应用于一对立体视图中。在这种情况下，左视图被编码为基本层，面右视图被编码为增强层。对于左视图，图像用 I，B 或 P 模式进行编码。对于 B 和 P 图像，用于预测的参考帧只来自于左视图。对于右视图，图像用 P 或 B 模式进行编码，在 P 模式中，图像用左视图中对应的帧作为参考帧进行预测编码。在 B 帧模式中，一个参考帧来自右视图中前面的一帧，而另一参考帧是左视图中对应的帧。后一种情况所用的运动矢量实际上是位差矢量，并且预测过程是位差补偿预测。

我们记得，对于双向预测，根据哪一个提供最佳预测，可以使用来自参考帧的预测块或使用二者的平均。通常，同一视图内的相继帧之间的相似性高于左右视图中对应帧之间的相似性。基于块的位差模型的不足进一步限制了用 DCP 可获得的精度。由于这些因素，一般选择 MCP 而不选择 DCP，使得多视类的总体性能只稍微优于同时联播，在同时联播中每个视图是用

MPEG-2 主类单独编码的[8]。多视类与同时联播的仿真比较结果可在参考文献[13]中找到。注意,对于同时联播和 MPEG-2 多视类,一般一个视图的编码质量大大低于另一个参考视图的编码质量。这是受到这一事实的启发:立体序列的接收质量很大程度上是由一个具有较高质量的视图决定的(见 12.1.2 节)。

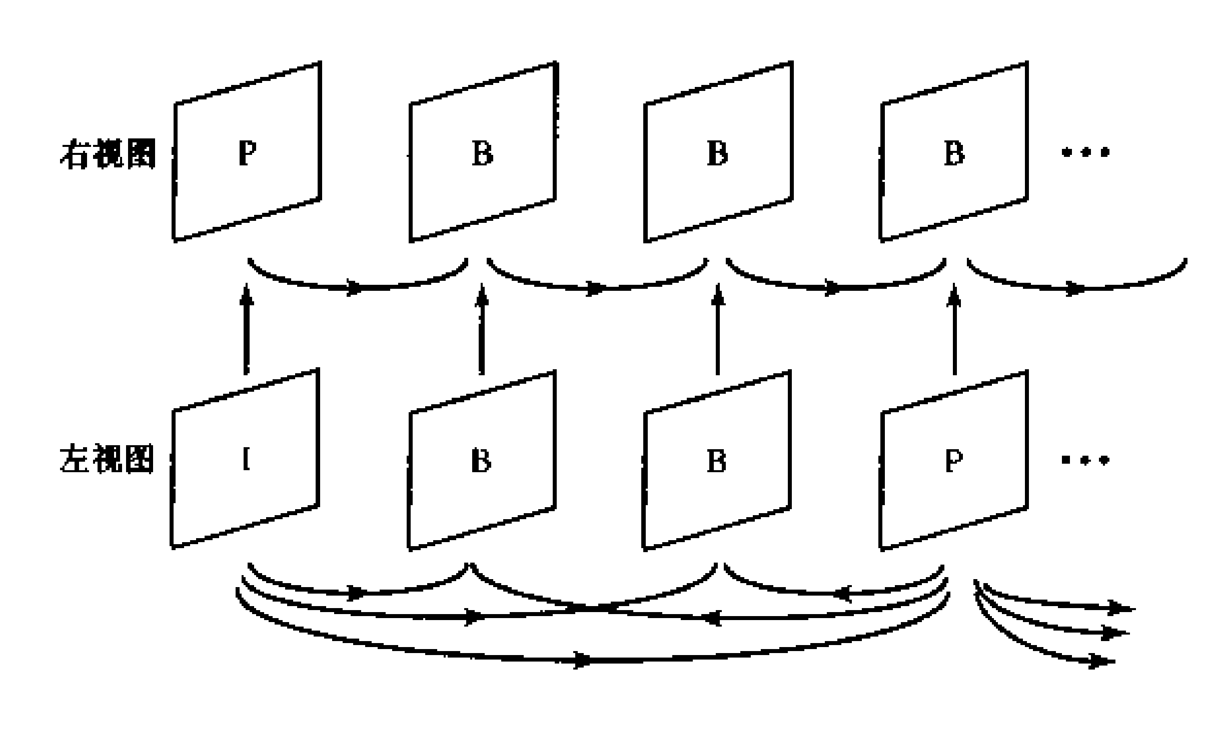
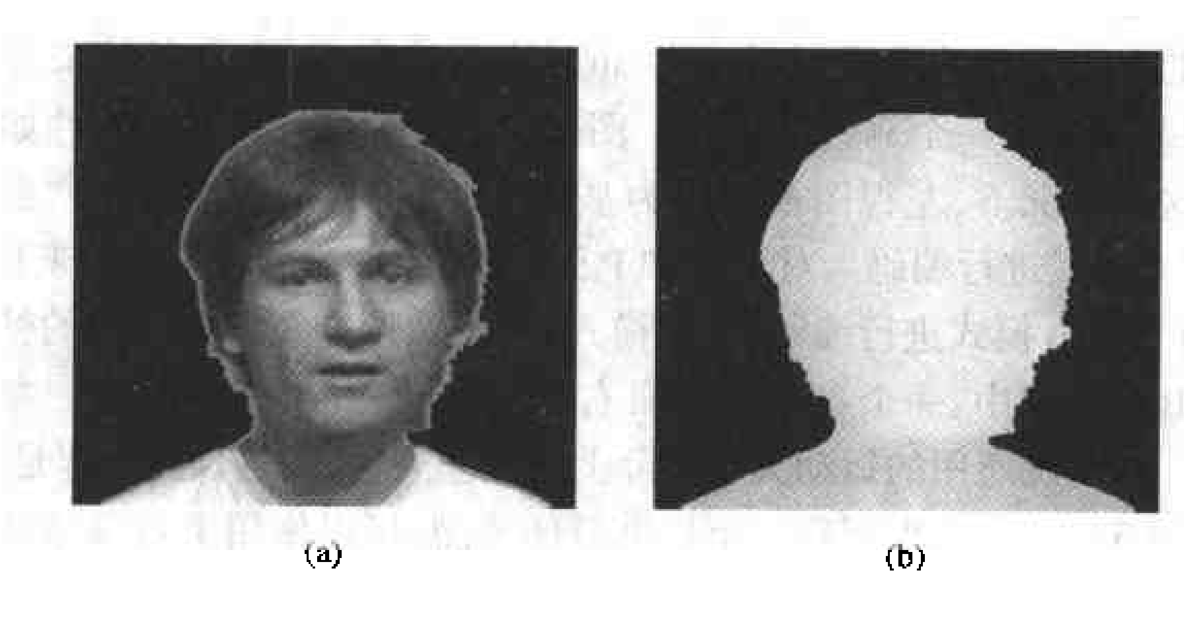


图 12.16 MPEG-2 的多视类

12.5.2 多视序列的不完全三维表示

在多视序列中,同一个物体区域经常是在多个视图中可见的。如果我们只在一个视图(提供最高分辨率表示的视图)中编码该区域的纹理图和该视图与其他视图之间的位差信息,那么我们可以用 DCP 相当好地重建它在其他视图中的图像。对于那些不要求非常高质量地重建每个视图的应用系统,不需要编码预测图像与原始图像之间的残差。代替分别编码每个区域的纹理图,可以把所有不同区域的纹理图放到一个扩充的图像中,并把这幅图像作为单个纹理图像对待。这个扩充的纹理图具有与每个区域有关的区域段和位差信息,组成了由 Ohm 和 Müller[9,8]提出的不完全的三维(I3D)表示。图 12.17 给出了序列"人"的 I3D 表示的例子。在这种情况下,左边和右边的脸纹理是分别由左视图和右视图获得的。把编码原来的两个视图换成编码 I3D 表示,这样可带来比特率的明显节省。用编码的位差信息,我们也可以内插中间视图。



(a) (b)

图 12.17 序列"人"的不完全三维表示:(a)扩充的纹理表面,(b)位差图。原始的左右图像见图 12.15

12.5.3 混合分辨率编码

如12.1.2节中所述,可以相当大地降低两个图像中任意一个的分辨率(用目前的电视标准,直到接近原始图像截止频率的一半),当图像短时间显示时,这不会引起清晰度的主观印象的刺激或变坏[11]。基于HVS的这个不对称性质,没有必要为立体对中的两个序列提供相同的空间和时间分辨率。在混合分辨率编码中[12,1,15],一个序列(例如左序列)是以最高可达到的空间和时间分辨率编码的,而另一个序列首先在空间和时间上被下采样,然后再编码。已经发现[15],可以对一个视图在水平和垂直方向都进行两倍的下采样,同时不会引起观看者感觉深度能力的下降。

12.5.4 基于物体的三维编码

这些方法比基于DCP和MCP的编码器又前进了一步。不必为进行MCP和DCP导出二维运动和位差,三维结构和运动参数是通过立体的或多重的视图估计的。每个物体的结构、运动和表面纹理(彩色)被编码,而不是编码各个图像帧。在解码器中,基于结构、运动和纹理信息合成所期望的视图。这样的方法比二维MCP/DCP联合的方法更有潜力。首先,以比二维运动和位差更直接、更简单的形式物理地约束三维运动和结构参数;适当地使用这些约束将导致更精确的三维运动和结构的估计。事实上,精确的三维运动估计应该得到更精确的二维运动,从而可降低表示每个视频所需要的比特数。其次,用从立体对中导出的三维信息,可生成任意的中间视图。这个特征在许多应用系统中都是所期望的——例如虚拟现实,它要求三维世界的连续视图,但这过于昂贵而难以获取或传输。最后,编码的三维信息能实现成像物体或场景的控制(改变视角、注释、动画等),这是交互式多媒体通信的重要特征。

这种编码方案的一般结构如图12.18所示。基于输入的立体视图,编码器首先进行物体分割和物体级的运动和结构估计。如12.3.6节所述,这三个任务是相互关联的,一般利用一个迭代程序得到最终解。每个物体是由线框描述的,从而运动和结构信息可由初始节点位置和节点位移矢量描述。对于每个物体,参考纹理图也是从两个视图中抽取出来的,这与12.5.2节描述的I3D表示一样。基于分割图以及结构和运动参数,我们便可通过参考纹理图合成左右视图,并且,如果需要的话,可以编码合成的残差图像。除了使用由给定视图构造的参考纹理图外,也可以选择输入视图中的一个作为参考纹理图,并合成另一个视图。这个合成可以用参考纹理图的位差补偿纹理扭曲实现(见12.4节)。需要编码的参数包括分割图、结构和运动信息、参考纹理图以及合成残差图像。类似的方法可应用于编码多视序列。符合这个一般结构的编码器可在参考文献[18,21]中找到。

基于物体的三维编码器的成功关键取决于它的三维结构和运动估计的精度。如果不知道在成像场景中有什么和有多少物体,试图由一般的立体序列恢复整个三维信息是一个非常困难的工作,甚至有时是不可能的。尽管对这个问题的研究已经付出了相当大的努力,并已开发了许多很有前途的方法,但现有的解决方案在它们的健壮性和精度上仍是有限的,并且需要强度很高的计算。由于有这些问题,基于物体的三维编码器的执行仍需要用非基于物体的编码器顺利地完成。

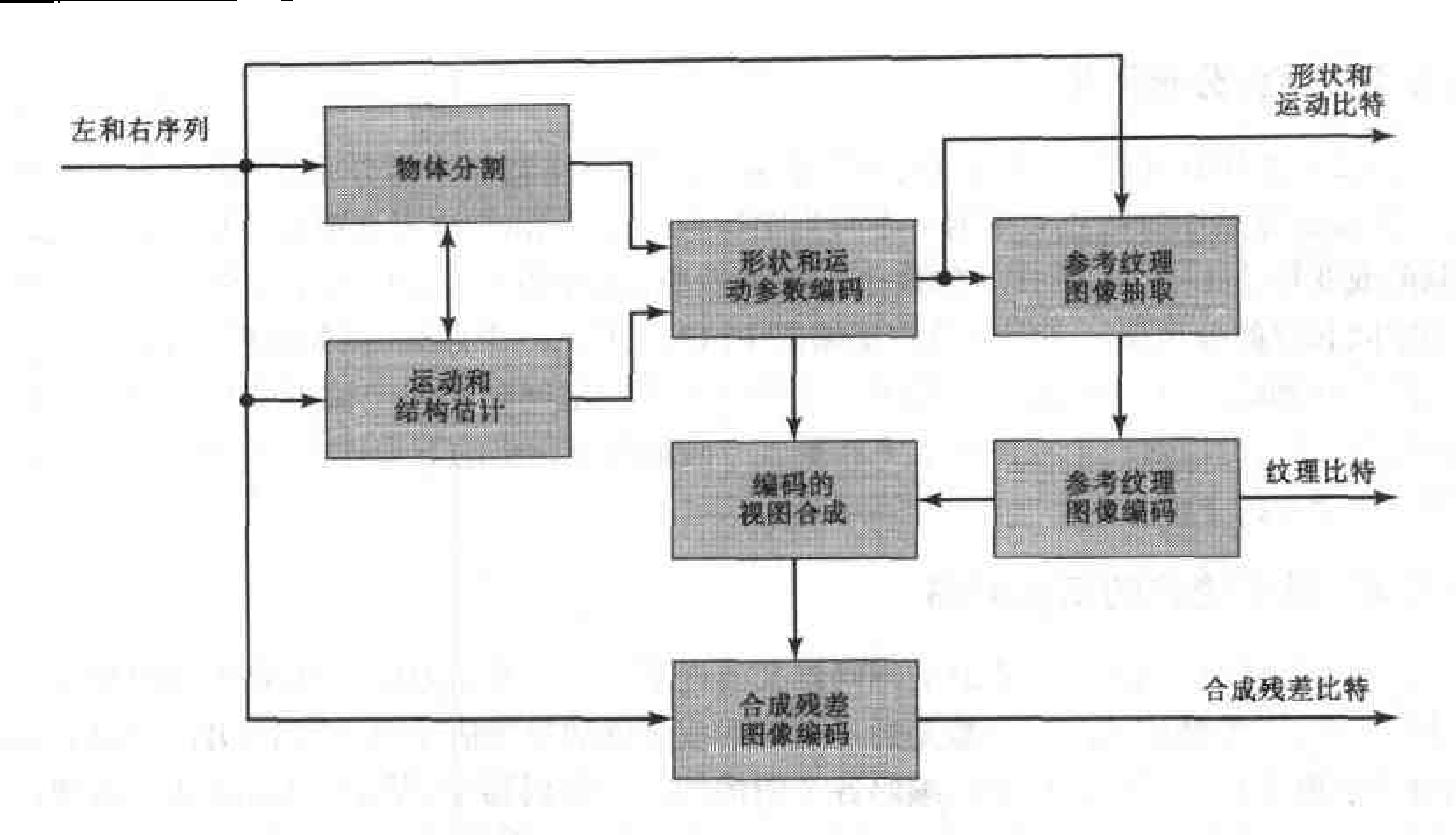


图 12.18 基于物体的一般体视序列编码结构

12.5.5 基于模型的三维编码

上述讨论试图从观察到的立体序列自动地导出场景中物体的三维结构。由于在成像场景中可能存在非常复杂的物体成分,所以这是个非常困难的问题。当场景中只有少数物体且物体的三维模型符合一些已知的结构时,困难要小一些。在这种情况下,可以为每个潜在的物体构建通用的模型。例如,在电视会议应用系统中,成像的场景一般由一个或几个头肩型的物体组成。编码器可使用预先设计的通用的脸和身体模型。结构估计问题就简化为估计通用模型需要修改之处,以便与采集的图像相匹配。一旦通用模型与成像的物体相适应,则只有运动信息(节点运动)必须被估计和编码。这种方法类似于前面第 10 章中所讨论的基于知识的编码。与为单视序列所开发的方法比较,立体或更多视图的可用性有助于提高模型自适应精度。

12.6 小结

深度感觉(12.1 节)

- 人类的深度感觉是通过两只眼睛将具有位移的视图融合在一起实现的,这种能力称为立体感。
- 眼睛对空间和时间的深度变化不如对比度变化敏感。因此,每个视图以标准单视序列的分辨率采集并显示的一个立体系统,可正确地显示深度。
- 事实上,两个视图不必用同样高的空间和时间分辨率显示。一个立体序列所感觉的质量取决于其中较好的视图。在立体序列压缩中已经利用了这个性质。

立体成像原理(12.2 节)

- 对于立体成像,一个三维点在两个不同视图中的像是相互位移的,正是这两个位置之间

的位差使得能够恢复该三维点的深度信息。

- 深度与位差之间最一般的关系由公式(12.2.5)给出。对于最流行的平行摄像机配置，它简化为著名的位差与深度之间的反比关系(公式(12.2.8))。对于任意的摄像机配置，同一个物体点的左右图像位置之间的关系由外极几何所描述(图 12.7；公式(12.2.13))。

位差估计(12.3 节)

- 位差估计类似于运动估计(由于必须在两幅不同的图像中寻找对应的点)，因此可用类似的方法实现，包括基于块的方法和基于网格的方法。
- 与运动估计相比较，位差估计可以借助于施加物理约束。最重要的是外极约束，这个约束是，对于右图像中的任意点，它在左图像中的对应点必须落在外极线上。对于平行摄像机设置，这意味着两幅图像中的对应点是在同一条水平线上。
- 位差估计是由立体序列恢复成像物体的三维结构和运动的基本步骤。为了实现这一目标，位差估计必须与物体分割、使用线框的三维模型化和遮挡区域的删除与处理等工作联合进行。

中间视图合成(12.4 节)

- 在虚拟现实显示中需要视图合成。在先进的多视序列编码系统中，它也是一个很重要的部分。
- 需要位差补偿内插以获得满意的结果。

立体和多视序列编码(12.5 节)

- 一般有两种类型的方法。第一种是基于波形的：它联合位差补偿预测和运动补偿预测，对预测残差图像以及位差和运动矢量进行编码。
- 第二种是基于物体的：它试图恢复成像物体的三维结构和运动，并直接编码其信息。对于结构简单的场景，第二种方法具有达到非常高压缩比的潜在能力。它也易于在解码器中合成任意的中间视图。
- 在 12.5.4 节和 12.5.5 节中所述的基于物体的和基于模型的方法，可认为是第 10 章中所讨论的为编码单视序列而开发的基于物体和基于知识的编码器的扩展。

12.7 习题

12.1 立体成像的平行摄像机配置与会聚摄像机配置的优缺点各是什么？

12.2 导出会聚摄像机配置中的水平和垂直位差值 $d_x(x_r, y_r)$，$d_y(x_r, y_r)$。说明对于 $X=0, Y=0$，垂直位差为零。同时导出由 $d_x(x_r, y_r)$，$d_y(x_r, y_r)$确定三维位置的方法。

12.3 对于(a)平行摄像机配置和(b)会聚摄像机配置，导出基本矩阵[**F**]。

12.4 假设是一个平行摄影机配置，对于弯曲的表面 $Z = aX + bY + cXY + d$，位差函数的参数化的形式是什么？

12.5 当两台平行的摄像机之间的基线较大时，水平位差很大。在这种情况下，最好在应用基于块的或基于网格的位差估计前，首先估计全局的水平位差值。这个全局位差可通过寻找移位后产生最小误差的两幅图像之间的全局水平位移来求得。写出一个估计立体对之间全局位差的程序。

12.6 在一个平行成像几何情况下（只须搜索水平位差）实现（用 C 或 MATLAB）位差估计的分层块匹配算法。评价它在立体图像对上的性能。应该首先估计全局位差，然后对全局位差补偿后的图像应用块匹配算法。

12.7 在平行成像几何情况下实现位差估计的基于网格的方法。为简化起见，在基准视图上采用规则的正方形网格。对于相同的立体对，把它的性能与块匹配算法进行比较。

12.8 实现在参考文献[10]中所描述的位差估计的动态编程方案。评价它在立体图像对上的性能。把它的性能与块匹配算法以及基于网格的方法进行比较。

12.9 在由平行摄像机获得的立体对的视图之间生成一个中间视图。比较两种方法：线性内插和采用基于网格方法的位差补偿内插（12.4 节）。理想情况下，应该用基于网格的方法估计位差值。如果选择用较简单的基于块的方法，那么可假设为每个块找到的位差是块中心的位差，并且所有的块中心是参考视图的网格中的节点。

12.10 用基于块的位差补偿实现一个立体图像编码器。用 DCT 方法（9.1.7 节）编码一幅图像（参考视图）。对于另一幅图像，对于每个块，用基于块的位差求它的预测，然后用 DCT 方法编码残差块。

12.11 用基于块的位差补偿和运动补偿实现一个立体视频编码器。用混合编码方法（9.3.1 节）编码一个视图（参考视图）。为简化起见，只用单向预测。对于另一个视图，对于帧中的每个块，用由这个视图中的前一帧进行的运动补偿和由参考视图中的同一帧进行的位差补偿来估计它。可以用两种预测的平均或用其中的一个预测，这取决于预测误差。然后用 DCT 方法编码预测误差。

12.8 文献目录

[1] Dinstein, I., et al. Compression of stereo images and the evaluation of its effects on 3-D perception. *SPIE Conf. Applications of Digital Image Processing* XII (1989), SPIE-1153:522–30.

[2] Falkenhagen, L. Depth estimation from stereoscopic image pairs assuming piecewise continuous surfaces. In *European Workshop on Combined Real and Synthetic Image Processing for Broadcast and Video Production*, 115–27. Hamburg, November 1994.

[3] Faugeras, O. *Three-Dimensional Computer Vision—A Geometric Viewpoint*. Cambridge, MA: MIT Press, 1993.

[4] ISO/IEC. IS 13818-2 AMD3: MPEG-2 multiview profile, Sept. 1996.

[5] Koch, R. Dynamic 3-D scene analysis through synthesis feedback control. *IEEE Trans. Pattern Anal. Machine Intell.* (1993), 15:556–68.

[6] Kopernik, A., R. Sand, and B. Choquet. The future of three-dimensional TV. In E. Dubois and L. Chiariglione. eds., *Signal Processing of HDTV.* Vol. IV, 17–29. Amsterdam: Elsevier Science, 1993.

[7] Malassiotis, S., and M. G. Strintzis. Model-based joint motion and structure estimation from stereo images. *Computer Vision, Graphics, and Image Understanding* (Jan. 1997), 65(1):79–94.

[8] Ohm, J.-R. Encoding and reconstruction of multiview video objects. *IEEE Signal Processing Magazine* (May 1999), 16(3):47–54.

[9] Ohm, J.-R., and K. Müller. Incomplete 3-D—multiview representation of video objects. *IEEE Trans. Circuits Syst. for Video Technology* (March 1999), 9(2):389–400.

[10] Ohta, Y., and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Trans. Pattern Anal. Machine Intell.* (1985), 7:139–45.

[11] Pastoor, S. 3-D-television: A survey of recent research results on subjective requirements. *Signal Processing: Image Commun.* (1991), 4:21–32.

[12] Perkins, M. Data compression of stereopairs. *IEEE Trans. Comm.* (1992), 4:684–96.

[13] Puri, A., R. V. Kollarits, and B. G. Haskell. Stereoscopic video compression using temporal scalability. *SPIE Conf. Visual Commun. Image Processing (VCIP95),* SPIE-2501:745–56.

[14] Ross, J., and J. H. Hogeben. Short-term memory in stereopsis. *Vision Res.* (1974), 14:1195–201.

[15] Sethuraman, S., M. W. Siegel, and A. G. Jordan. A multiresolution framework for stereoscopic image sequence compression. *IEEE Int. Conf. Image Processing (ICIP94),* 2:361–65.

[16] Sexton, I., and P. Surman. Stereoscopic and autostereoscopic display systems—an in-depth review of past, present, and future technologies. *IEEE Signal Processing Magazine* (May 1999), 16(3):85–99.

[17] Stereographics, Inc. *Perceiving Stereoscopic Images.* (http://www.stereographics.com/html/chapter_2.html).

[18] Strintzis, M. G., and S. Malassiotis. Object-based coding of stereoscopic and 3-D image sequences. *IEEE Signal Processing Magazine* (May 1999), 16(3):14–28.

[19] Tsai, R. Y. A versatile camera calibration technique for high-accuracy 3-D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation* (Aug. 1987), 3:323–44.

[20] Tzovaras, D., M. S. Strintzis, and H. Sahinoglou. Evaluation of multiresolution techniques for motion and disparity estimation. *Signal Processing: Image Commun.* (June 1994), 6(1):59–67.

[21] Wang, R., and Y. Wang. Multiview video sequence analysis, compression, and virtual viewpoint synthesis. *IEEE Trans. Circuits Syst. for Video Technology* (April 2000), 10(3):397–410.

[22] Weng, J., T. S. Huang, and N. Ahuja. *Motion and Structure from Image Sequences.* New York: Springer-Verlag, 1993.

[23] Zhang, Z. Determining the epipolar geometry and its uncertainty: A review. Technical report No. 2927, INRIA, Sophia-Antipolis, France, July 1996. (Available from http://www-sop.inria.fr/robotvis/personnel/zzhang/publications.html)

第13章　视频压缩标准

数字视频通信是一个复杂的、计算强度大的过程，要求许多人接收来自不同信源的视频信号。目前主要存在三类用于数字视频通信的设备：

- 数字电视机或机顶盒，它主要是为接收来自不同内容提供商的视频信号而设计的。这些设备依赖单一固定的视频解码算法，这种算法是以硬件、硬件与可编程的精简指令集计算机(RISC)处理器的组合或数字信号处理器(DSP)实现的。目前，一旦硬件为用户开发出来就没有办法上载新的算法。
- 视频电话一般在DSP上实现，对视频编码和解码算法的某些计算复杂度高的部分(例如，DCT和运动估计)用硬件加速。通常，在特殊的视频电话中使用的算法集是不能替换的。
- 个人计算机是数字视频通信最灵活和最昂贵的平台。尽管具有高级奔腾Ⅲ处理器的PC机能解码DVD，但一般用操作系统预先安装软件以避免硬件和驱动器问题。流视频的视频解码器可通过自动软件下载和安装进行更新，如商业软件Real Player，Windows Media Player，Apple QuickTime和Microsoft Netmeeting，等等。

数字视频通信标准主要是为视频电视和视频电话开发的，从而使业界能向用户提供合理价位的有效带宽终端。13.1节描述了标准化组织、兼容性含义和视频编码标准应用系统。我们以ITU交互式视频通信的视频编码标准H.261和H.263(13.2节)开始实际标准的描述。在13.3节中，我们给出H.323和H.324标准，它们定义了音视频通信的多媒体终端。ISO下属的运动图像专家组(MPEG)定义了娱乐和数字电视的MPEG-1(13.4节)和MPEG-2(13.5节)标准。MPEG-4(13.6节)是第一个不仅对音频和视频通信而且对用于娱乐和交互式多媒体业务的图形进行标准化的国际标准。所有的标准都描述比特流的语法和语义。13.7节给出了一个被H.261、H.263和MPEG-1、MPEG-2和MPEG-4所采用的比特流结构的概述。最后，我们简单地介绍正在进行的MPEG-7标准化工作，它意在对描述音视频文件内容的接口进行标准化。

13.1　标准化

开发一种国际标准需要来自不同国家的许多同行的合作，并需要一个能支持标准化过程和实施标准的组织。在13.1.1节中，我们将介绍如ITU和ISO这样的组织。在13.1.2节中，定义兼容性的含义。13.1.3节简单叙述标准化组织的工作。在13.1.4节中，列出了视频通信的应用。

13.1.1　标准化组织

如果我们想要不同厂家生产的各种终端交换或接收来自公共信源(如电视广播台)的信息，就需要制定一种标准。标准化组织面向电信行业创建了ITU(国际电信联盟)，面向贸易产

业创建了 ISO(国际标准化组织)。

ITU　电信行业在设立国际标准方面已有很长的历史[7]。最初是 19 世纪的电子电报，当时电报线路不是跨国界的，因为每个国家使用不同的系统，并且每个国家有它们自己的电报码以保护它的军事和政治消息的秘密。消息在通过邻国的电报网络传输之前，必须在边境转录、翻译和移交。1865 年 5 月签署了第一份国际电报协定，它协调了当时正在使用的几种不同的系统，这一事件标志着国际电信联盟 (ITU; www.itu.int) 的诞生。

随着电话的发明以及随之而来的电话的推广，电信联盟在 1885 年开始起草电话国际规则。1906 年签署了第一个国际无线电电报协定。随后，成立了几个建立国际标准的委员会，包括 1924 年成立的国际电话咨询委员会(CCIF)，1925 年成立的国际电报咨询委员会(CCIT)，以及 1927 年成立的国际无线电咨询委员会(CCIR)。1927 年电报联盟把不同频段分配给当时存在的各种无线电业务。在 1934 年，1865 年成立的国际电报大会和 1906 年成立的国际无线电电报大会合并为 ITU。1956 年，CCIT 和 CCIF 合并，创建了国际电话和电报咨询委员会(CCITT)。1989 年 CCITT 发布了第一个数字视频编码标准：CCITT 建议 H.261[41]，这个标准今天仍是适用的。1992 年，ITU 进行了自身改造，重新命名 CCIR 为 ITU-R，命名 CCITT 为 ITU-T。从而，CCITT 的标准现在称为 ITU-T 建议(例如，CCITT H.261 现在称为 ITU-T H.261)。

图 13.1 示出了 ITU 的组织结构，详细说明了与数字视频通信有关的部分。ITU-T 由研究组组成，研究组 16(SG16)负责多媒体。SG16 把它的工作分为不同的工作组(WP)，每个工作组处理几个议题。SG16 在 2001 年的工作议题是议题 15 (先进视频编码)和议题 19 (对现有比特率低于 16 kbps 的 ITU-T 语音编码标准的扩展)，议题 15 发展了视频编码标准 ITU-R 建议 H.261 和 H.263[50]，议题 19 发展了语音编码标准，如 ITU-R 建议 G.711[36]，G.722[37]以及 G.728[39]。议题号倾向于每四年变一次。

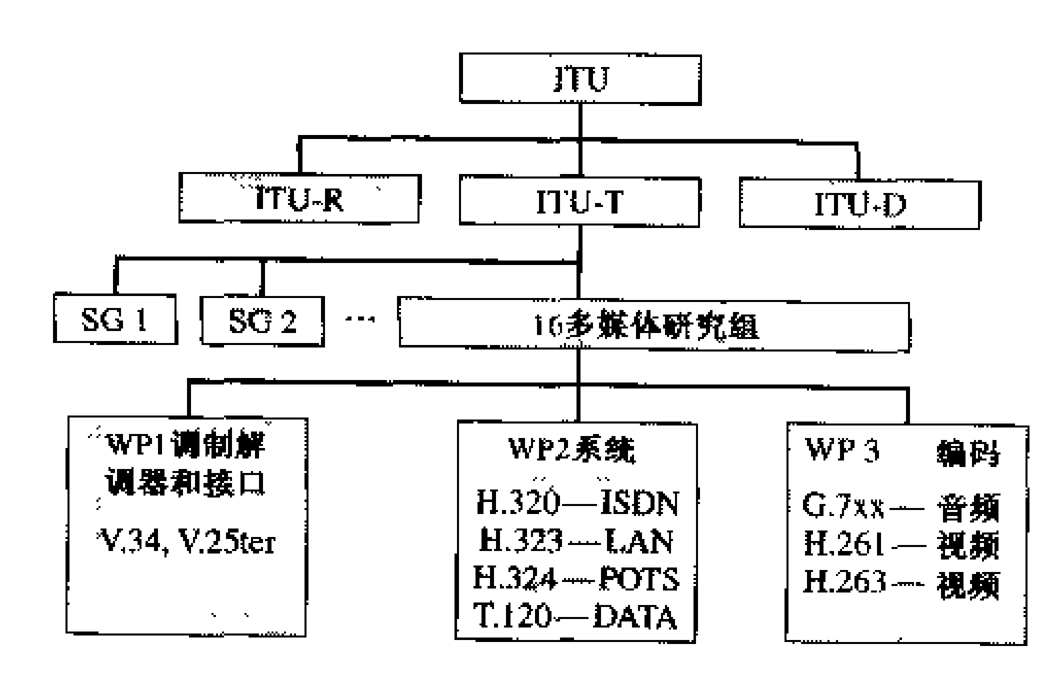


图 13.1　ITU 的组织以及它与数字视频通信有关的小组。工作组被组织到定义标准的议题中

ITU 是一个国际组织，通过由它的成员国签署的条约建立。成员国相对于它们的自主权考虑 ITU 的规定。相应地，ITU 的任何建议(即标准)都必须得到成员国的一致同意。因此，ITU 标准化过程时常不能跟上现代技术的发展。有时，当不能达成一致的决议时，ITU 就推荐地区标准(如在美国和欧洲分别用 7 bit 和 8 bit 表示数字语音)。到移动电话被关注时，ITU 并没有起到领导的作用。在美国甚至没有国家移动电话标准，使得每个经营者可以自由地选择他们自己喜欢的标准。这与欧洲所采取的方法形成对比，在欧洲 GSM 标准已经获得成功，并

扩展到世界各地，包括美国。通过 UMTS（所谓第三代移动标准），ITU－T 正重新扮演原来的全球移动通信标准开发者的角色。

ISO 建立国际标准的需求随着贸易的发展而发展[7]。国际电工委员会(IEC)成立于1906年，为了准备并发布所有电气、电子以及相关技术的国际标准。IEC 目前负责如"接收机"、音频和视频记录系统以及目前都归类于 TC 100(音频、视频和多媒体系统和设备)的音视频设备这样一些通信工具的标准。其他领域(特别是机械工程)的国际标准化是由建于1926年的国家标准化协会国际联盟(ISA)关注的。ISA 的活动于1942年停止。但是在1947年一个新的称为国际标准化组织(ISO;www.iso.ch)的国际组织开始工作，声称其目标是"要推动工业标准的国际协调和统一"。所有与计算机相关的活动目前都由关于信息技术的 ISO/IEC 联合技术委员会1(JTC 1)负责。这个委员会规模已很大；大约30%的 ISO 和 IEC 标准工作都在 JTC 1 中完成。

分委员会 SC 24(计算机图形和图像处理)和 SC 29(音频、图片、多媒体和超媒体信息编码)是与多媒体通信有关的。SC 24 定义计算机图形标准，例如 VRML，而 SC 29 开发了众所周知的音视频通信标准 MPEG-1，MPEG-2 和 MPEG-4(图 13.2)。这个标准是在来自工业、研究院和大学的200到400名代表参加的会议上制定的。

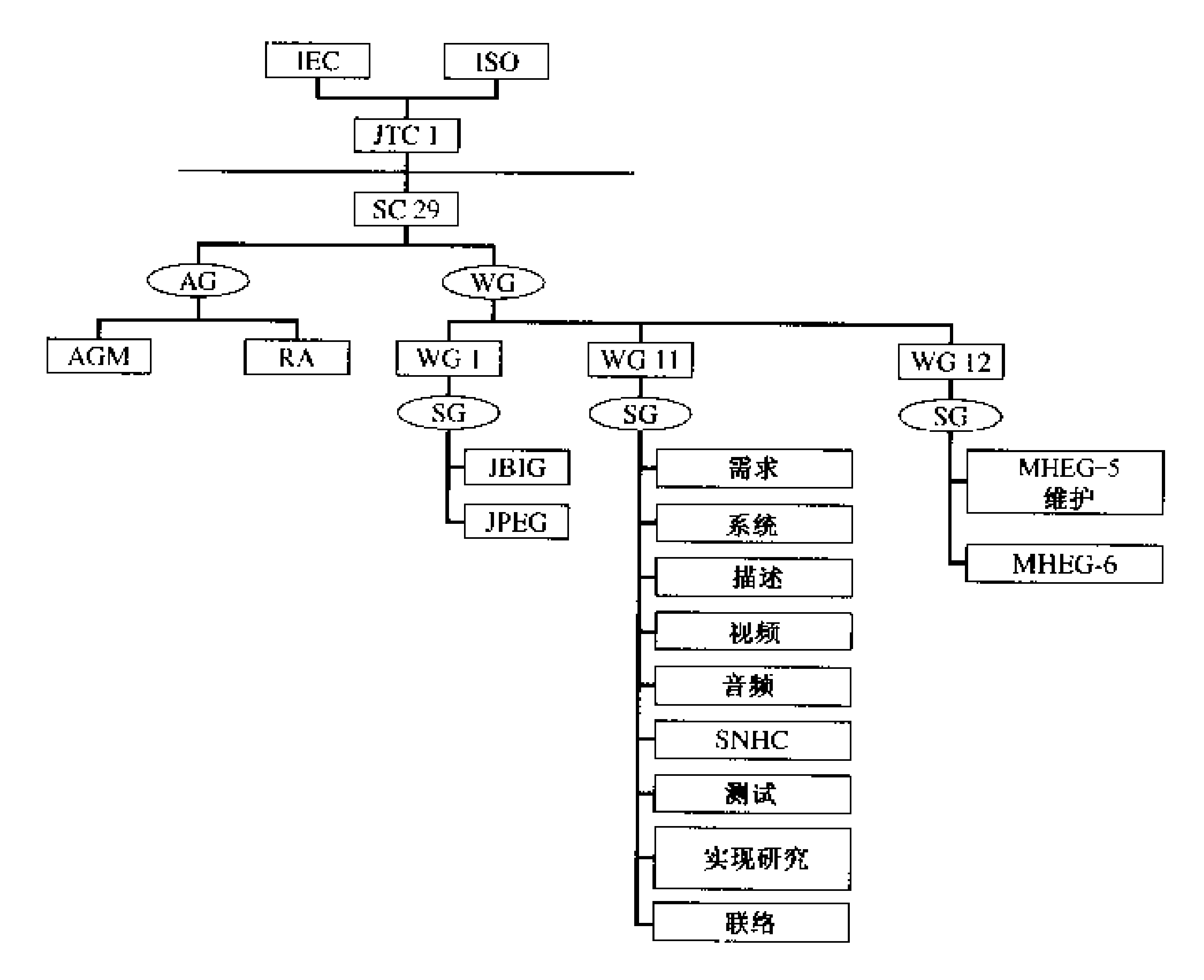


图 13.2 音视频通信标准如 MPEG-1，MPEG-2 和 MPEG-4 是由 ISO/IECJTC1下属的第29分委员会的第11工作组制订的

1947年以来，ISO 已成为联合国的一个代理机构，ISO 和 IEC 都是非政府的非赢利性质的组织。与 ITU 类似，为公布一个标准，ISO 需要一致同意。ISO 有时也不能建立真正的国际标准，如在数字电视方面所见到的。尽管相同的视频解码(MPEG-2 视频)在全球广泛使用，然而音频表示在美国和欧洲是不同的。

ISO 和 ITU 都一直在与工业竞争。尽管 ISO 和 ITU 在定义广泛应用的音频和视频编码标准方面是很成功的，但在定义因特网上多媒体信号传输方面却不太成功。这方面目前由因特网工程任务组(IETF; www.ietf.org)负责；IETF 是一个大而开放的网络设计者、经营者、厂商以及涉及因特网结构发展和平稳运行的研究人员的国际团体。IETF 向任何有兴趣的个体开放。其他的实际上的标准(例如 JAVA)都是由一个或几个公司制定的，因此限制了局外和后发企业进入该技术。

13.1.2　对成功标准的要求

国际标准的开发允许由不同厂商提供的通信设备相互配合工作。因此，下面这些要求使得音视频通信设备在市场上能成功地推广。

1. 创新：为了使一个标准能与其他已被采用或已被广泛接受的工业标准相区别，它必须提供相当数量的创新。对视频编码来说，创新就意味着该标准提供新的功能，例如广播质量的隔行扫描数字视频、CD－ROM 视频或改善的压缩。如果一个新的标准仅有的特点是较好的压缩，那么在它的引入取得商业意义之前，这个标准至少应该对用户和非专家提供一个可见的改善。这个改善通常解释为在一般可接受的图像质量水平上，压缩视频有 3 dB PSNR 增益。
2. 竞争：标准不应该阻止制造商之间的竞争。因此，标准的规范必须对任何人都是开放的和可利用的。免费的编解码软件也有助于促进一个标准。而且，标准应该只定义比特流的语法和语义，即标准定义解码器如何工作。比特流生成不进行标准化。尽管比特流语法和语义的开发需要编码器和解码器，但标准不定义编码器。因此，依从标准的终端的制造商可以不只在价格上，也可以在附加性能上进行竞争，例如解码媒体的后处理以及更重要的编码器性能。在视频编码中，性能的主要区别来自于特殊的运动估计、场景变化处理、码率控制以及最优比特分配。
3. 独立于传输和存储介质：一个内容供应商应该能独立于网络或存储介质数字地传输或存储编码的内容。这种要求的结果是，音频和视频标准用于编码音视频信息，然后系统标准用于把音频和视频比特流格式化成一种适合于所选择网络或存储介质的格式。系统标准规定了打包、复用以及传送音频和视频比特流的包头语法。传输介质和媒体编码的分离通常对特定的应用系统造成额外开销。
4. 向前兼容性：一个新的标准应该能理解先前标准的比特流。例如，一个新的视频编码标准，如 H.263[50]，应该能解码按照以前的视频编码标准 H.261[41]编码的比特流。向前兼容性确保新产品能逐渐进入市场。只有当依从最新标准的终端能够通信时，最新标准的新特性才会被采用。否则，终端按照以前的标准相互配合工作。
5. 向后兼容性：如果旧标准能解码新标准的比特流，则新标准是对旧标准向后兼容的。一个非常重要的向后兼容标准是模拟彩色电视的引入。黑白接收机能够接收彩色电视信号并能显示信号的一个稍微降质的黑白版本。今天的数字视频标准的向后兼容性可以通过定义比特流中解码器能忽略的保留比特实现。新标准可利用旧标准的这些保留比特传输额外的信息。这样，旧的终端将能够按照新标准解码比特流。而且，他们知道比特流中那些符合旧标准的部分。向后兼容性对新标准能够达到的性能改善可能会施加苛刻的限制。因此，在新标准中不一定实现向后兼容性。

6. 向上兼容性：一种新的接收机应该能够解码为前一代或较便宜的一代相似接收机所制定的比特流。如果现存的标准被扩展，向上兼容性是重要的。一种新的 HDTV 机应该能够接收标准清晰度的电视信号，因为这两种接收机都采用相同的 MPEG-2 标准[19]。
7. 向下兼容性：一种旧的接收机能够接收并解码新一代接收机的比特流。如果现存的标准被扩展，向下兼容性是重要的。这可以通过只解码部分比特流实现，如果新的比特流是可分级的(第 11 章)，这是容易做到的。

显然，为了标准的广泛采纳，不是以上所有的要求都是重要的。我们认为重要性依次为创新、竞争以及向前兼容。对于像电视机顶盒或移动电话这样的设备，兼容性是最重要的，因为它们不易升级。然而，当今出现的任何多媒体 PC 机都安装了十个以上的视频编解码软件，对于这类终端，视频和音频编码标准的兼容性就显得不太重要。

13.1.3 标准的发展过程

视频编码标准的发展分为三个阶段：竞争、集中和验证。图 13.3 示出了视频编码标准 H.261[41]的发展过程。竞争阶段始于 1984 年。标准于 1990 年 12 月发布，于 1993 年修订。

主题	1985	1986	1987	1988	1989	1990
$n\times384$ ($n=1\sim5$)	竞争 RM	集中 1 2 3 4	最优化验证			
$p\times64$ ($p=1\sim30$)				集中 RM 6 7 8		最优化验证 REC H.261
$m\times64$ ($m=1,2$)				竞争 RM 5		

图 13.3 H.261 标准化过程概述。起初，目标是为码率为 $n\times384$ kbps 和 $m\times64$ kbps 的视频编码制定两种标准。最后，ITU 决定了一个码率为 $p\times64$ kbps的标准

在竞争阶段，定义了标准的应用范围及需求。进一步，专家聚在一起证明他们的最好算法。通常，一旦定义了需求，标准化组织就发出一个征求建议通知，以便征求整个社会的进入。这个阶段的特点是独立地进行竞争性的实验。

集中阶段的目的是合作实验以达成编码方法的一致。这个过程由彻底评价对该标准的建议开始。当对标准的第一个公共框架达成一致时，考虑如编码效率、主观质量、实现复杂度以及兼容性等问题。在不同的实验室实现这个框架，并且精炼其描述，直到不同的实现达到相同结果。这个框架在不同的标准中有不同的名称。如 H.261 的参考模型(RM)、H.263 的近端测试模型(TMN)、MPEG-1 的模拟模型(SM)、MPEG-2 的测试模型(TM)、MPEG-4 的验证模型(VM)以及 H.26L 的远端测试模型(TML)。第一个框架版本实现后，研究人员提出改进意见，例如算法的新部分，或者已有部分的更好的参数。这些都相对于当前的框架进行评价。取得显著改进的建议被包括在下一个框架版本中，以作为进一步改进的新参考。反复进行这样的过程直到达到期望的性能水平。

在验证阶段，检查说明书是否有差错和歧义。生成一致性比特流和正确解码的视频序列。依从标准的解码器必须把每一个一致性比特流解码成正确的视频序列。

H.261的标准化过程可以作为一个典型的例子(图13.3)。在1985年,最初的目的是要开发一种比特率在384~1 920 kbps之间的视频编码标准。由于ISDN电话线路的使用,两年后开始了另一种64~128 kbps视频编码的标准化过程。1988年,两个标准化小组认识到可用一种算法进行64~1 920 kbps的视频编码。RM 6是第一个覆盖这个比特率范围的参考模型。技术结束于1989年,一年后,ITU正式采纳了H.261。

13.1.4 现代视频编码标准的应用

正如已提到的,存在几个主要的视频编码方面的积极性,它们导致了用于不同应用系统的一系列视频标准的出现[12]。

- 视频电视会议的视频编码导致了用于ISDN视频会议的ITU标准H.261[41],用于在模拟电话线路传输视频会议和连接到因特网的桌面和移动终端的H.263 [50],以及用于ATM/宽带视频会议的H.262/MPEG-2视频[43,17]。
- 用于在CD-ROM上存储电影以及其他消费视频应用系统中的视频编码,把大约1.2 Mbps分给视频编码,256 kbps分给音频编码,导致了最初的ISO MPEG-1标准[16]。如今,MPEG-1用于CD、卡拉OK机、一些数字便携式摄像机和因特网上的消费视频。一些数字卫星在MPEG-2发布前用MPEG-1播放电视信号。
- 用于广播和DVD数字视频存储的视频编码,大约2~15 Mbps码率分配给视频和音频编码,导致了ISO MPEG-2标准[19]以及由数字音视频委员会(DAVIC; www.davic.org)和DVD协会制定的DVD操作说明书[25]。这项工作扩展到了分配码率范围为15~400 Mbps的HDTV视频编码。应用系统包括卫星电视、有线电视、地面广播、视频编辑和存储。如今,MPEG-2视频被应用于每一个数字机顶盒。它已被选择为美国HDTV广播系统的视频解码器。
- 分离的音视频物体(包括自然的和合成的)的编码在ISO MPEG-4[22]中被标准化。目标应用系统是因特网视频、交互式视频、内容管理、专业视频、二维和三维计算机图形以及移动视频通信。

在以下各节中,我们将首先描述标准H.261,然后,将重点说明H.263与H.261的不同,并且比较它们的编码效率。然后,我们将讨论MPEG-1,MPEG-2和MPEG-4,也是集中讨论它们的区别。

13.2 采用H.261和H.263的视频电话

64 kbps的视频编码最早是在1979年的一次会议上论证的[57]。大约花费了十多年的时间来定义在这个码率下的商业上可行的视频编码标准。H.261标准发布于1990年,从而使视频会议使用1~30个ISDN信道。那时不同厂商的视频会议硬件都是可用的。如销售具有专利算法视频会议设备的PictureTel公司很快就提供了H.261作为可选项。后来,ITU开发了类似的标准H.263,它能使视频通信在模拟电话线路上传输。如今,H.263视频编码器和解码器软件被安装在每一个具有Windows操作系统的PC机上。

13.2.1 H.261 概述

图 13.4 示出了以 4:2:0 采样格式处理视频的 H.261 编码器方框图。它是一种具有运动补偿的基于块的混合编码器(9.3.1 节),它把图像分成 16×16 像素尺寸的宏块。一个宏块由 4 个亮度块和两个色度块(一个用于 Cr 分量,另一个用于 Cb 分量)组成。H.261 对每个块使用 8×8 DCT 以减少空间冗余度,使用 DPCM 环路以利用时间冗余度,并对宏块使用单向整数像素向前运动补偿(图 13.4 中的方框 P),以提高 DPCM 环路的性能。

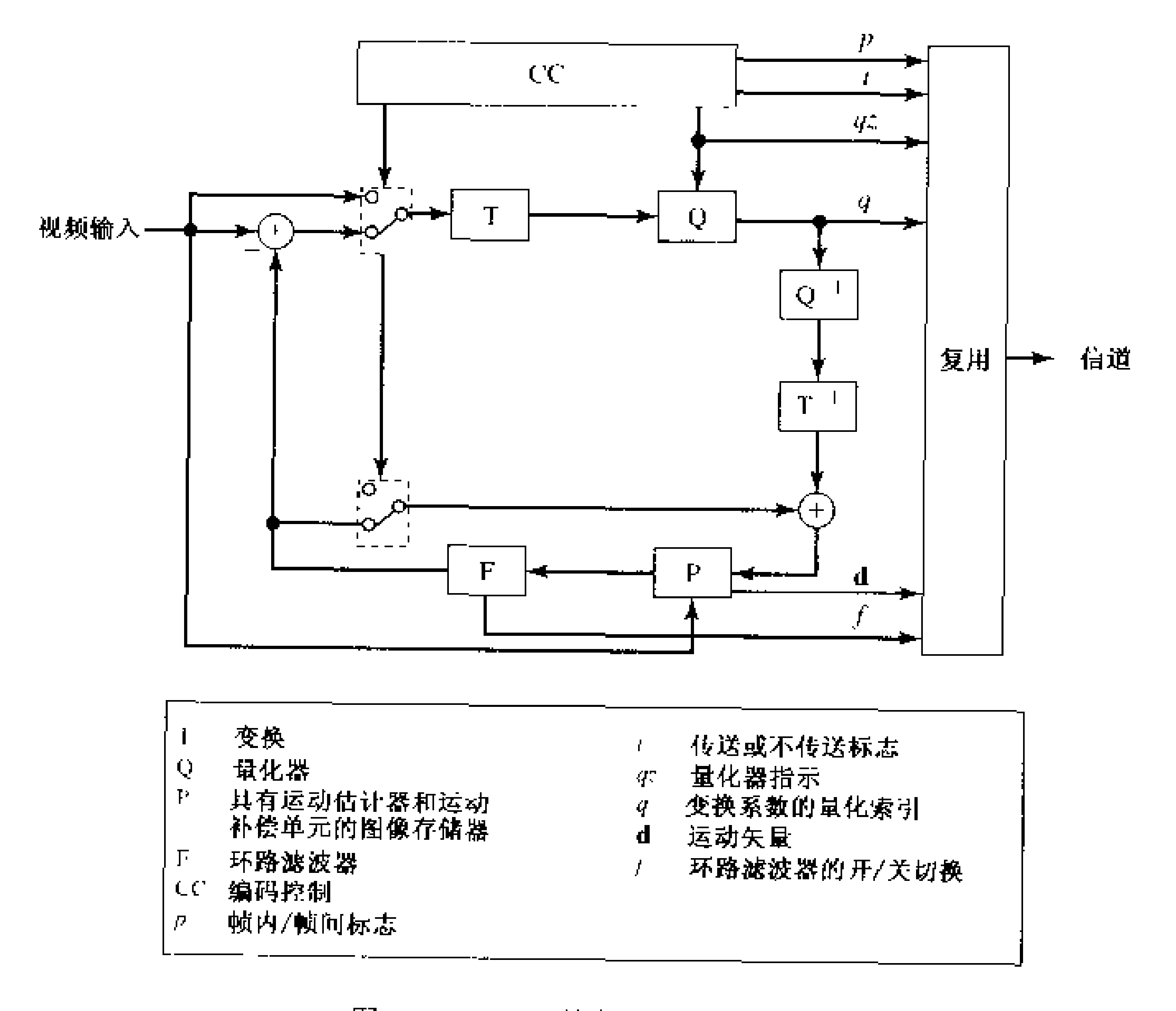


图 13.4 H.261 编码器的方框图[41]

可采用一种简单的二维环路滤波器(见 9.3.5 节)对运动补偿预测信号进行低通滤波(图 13.4 中的方框 F)。它一般会减少预测误差并降低预测图像的块效应。环路滤波器可分离为具有系数$\left[\frac{1}{4},\frac{1}{2},\frac{1}{4}\right]$的一维水平和垂直函数。H.261 对 DCT 系数采用两个量化器。用步长为 8 的均匀量化器量化帧内模式的 DC 系数,用步长为 2 至 62 的接近均匀的中间踏板量化器量化帧内模式和帧间模式的 AC 系数(图 13.5)。在 $-T$ 到 T 之间的输入称为死区,被量化为 0。除了死区外,步长是均匀的。这个死区避免对主要会引起编码噪声的许多小的 DCT 系数进行编码。

解码器主要传输每个编码宏块的两类信息:由预测误差信号变换产生的 DCT 系数(图 13.4 中的 q)和由运动估计器估计出的运动矢量(图 13.4 中的 d 和方框 P)。运动矢量的范围限制在 ±16 个像素内。通知解码器一个宏块以及它的块是否编码以及如何编码的控制数据称为宏块类型(MTYPE)和编码块模式(CBP)。表 13.1 示出了各种宏块类型及其相应的 VLC

码字。在帧内模式中，比特流包含每个块的变换系数。作为选择项，可以标示量化器步长(MQUANT) ± 2 的改变。在帧间模式中，编码器可选择只传送具有或不具有环路滤波的差分编码运动矢量(MVD)。可以选择传送 CBP，以便指定需要传递变换系数的块。由于标准不规定编码器，由编码器厂商确定一种有效的编码控制(图 13.4 中的 CC)来最优地选择 MTYPE、CBP、MQUANT、环路滤波器以及运动矢量(MV)[70]。作为粗略的原则，我们可以选择 MTYP，CBP 以及 MVD 使预测误差为最小。然而，因为传送 MV 花费额外的比特，所以只当使用 MV 比不用 MV 预测误差低很多时，才传送 MV。编码图像时量化器步长是变化的，从而图像编码所需比特数不需要比编码器在两个编码帧之间所能传送的比特数更多(在 9.3.3 节中讨论过编码模式和参数选择)。

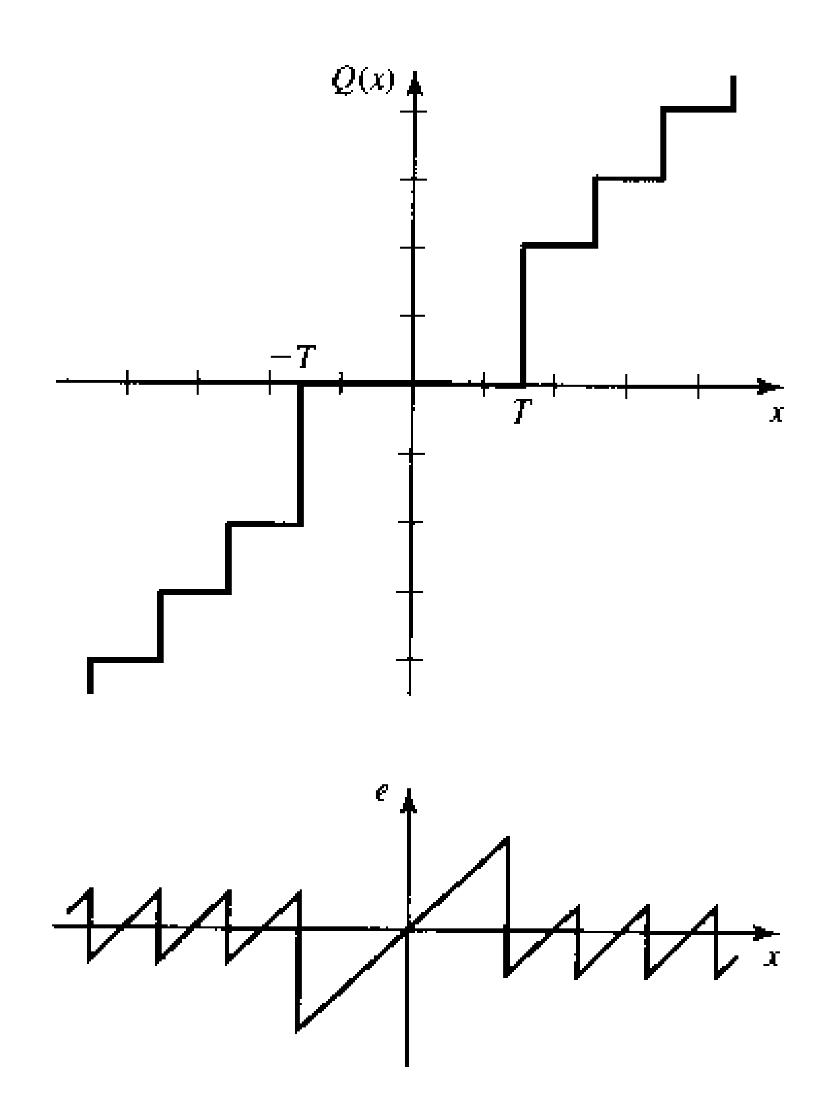


图 13.5　在 H.261 中使用的一个用于(除帧内模式的 DC 系数外)所有 DCT 系数的具有死区的中间踏板量化器。图的下部表示了输入幅度 x 与输出幅度 $Q(x)$ 之间的量化误差 $e = x - Q(x)$

表 13.1　宏块类型的 VLC 表①

预测	MQUANT	MVD	CBP	TCOEFF	VLC
帧内				x	0001
帧内	x			x	0000 001
帧间			x	x	1
帧间	x		x	x	0000 1
帧间 + MC		x			0000 0000 1
帧间 + MC		x	x	x	0000 0001
帧间 + MC	x	x	x	x	0000 0000 01
帧间 + MC + FIL		x			001
帧间 + MC + FIL		x	x	x	01
帧间 + MC + FIL	x	x	x	x	0000 01

①两个 MTYPE 用于帧内编码的宏块，8 个 MTYPE 用于帧间编码的宏块。"x"表示对该宏块所传送的语法元素[41]。

宏块内的大部分信息是用由测试序列的统计特性导出的可变长码进行编码的。二维 DCT 系数是用 9.1.7 节中讨论的游程编码方法编码的。具体地说，量化的 DCT 系数通过“Z”形扫描(图 9.8)转换成符号(图 9.9)。每个符号包括从最近的一个非零系数以来被量化为零的系数个数以及当前的非零系数的幅度。注意在图 9.9 所述的例子中，根据 JPEG 图像编码方法，DC 系数是个别地进行编码的。对于 H.261 和其他的视频编码标准，所有的 DCT 系数都转换成(游程，值)这样的符号。如果 DC 系数是非零的，那么第一个游程为零。这样，图 9.9 中的第一个编码符号应该是(0,5)。每个符号用 VLC 进行编码。在块的最后一个非零系数之后，编码器输出一个块结束(EOB)符号。

H.261 并未规定视频编码器的能力。然而，表 13.2 中列出了 H.261 解码器必须支持的图像格式。建立视频会议通话的几个标准也交换终端之间的视频能力。在 H.320 定义的最小级别上，一个解码器必须能以 7.5 Hz 的速率解码 QCIF 帧[46]。可选的能力级别定义为以 15 Hz 解码 CIF 帧[46]。最大的级别要求以 30 Hz(精确地说是 30000/1001 Hz)的速率解码 CIF 帧[46]。

13.2.2 H.263 简介

H.263 标准是基于 H.261 框架的。由于视频压缩技术的进步以及合理价格的高性能台式计算机的普及使用，ITU 决定在 H.263 标准中放入计算能力更强且效率更高的算法。H.263 的发展分为三个阶段。初始标准的技术工作于 1995 年 11 月完成。H.263 的扩展，别名为 H.263+，于 1997 年 9 月并入到标准中。第三个阶段的结果，别名为 H.263++，于 1999 年合并到标准中，并且于 2000 年 11 月正式通过。在这一节中我们集中讨论 1995 年的 H.263 与 H.261 的区别。我们也简单地描述 2000 年的 H.263。

H.263 基线(1995 年)与 H.261 H.263 包括一个基线解码器，该基线解码器具有任何 H.263 解码器必须实现的特性。另外还定义了可选特性。以下必须遵循的特性使 1995 年 11 月制定的 H.263 与 H.261 相区别[6,12]：

1. 半像素运动补偿：在物体运动需要高的空间分辨率以实现精确模拟的情况下，这个特性可大大提高运动补偿算法的预测能力。在非整数运动矢量的情况下，用双线性内插(简单平均)计算预测像素。运动矢量的编码是用三个邻近宏块的中值运动矢量作为对该矢量的每个分量的预测(图 13.6)。

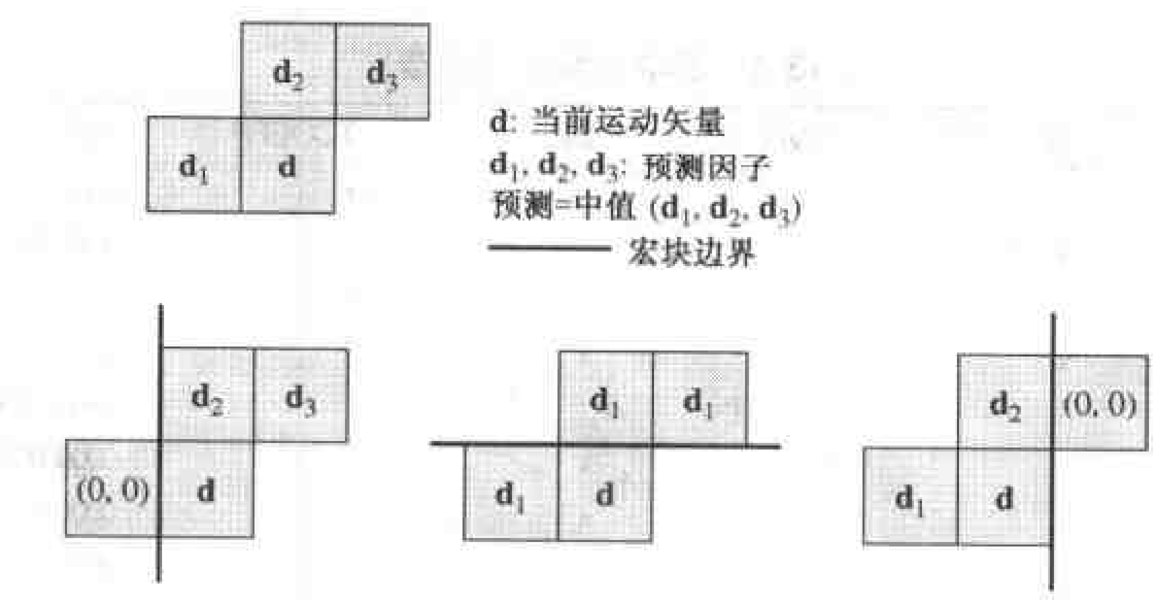


图 13.6 使用 d_1，d_2 和 d_3 中值的运动矢量预测。如果一个宏块在图像或组块的外面，则我们假设其运动矢量为零。如果有两个宏块在外面，则我们用剩下的运动矢量作为预测

2. 改善的可变长编码,包括三维 VLC 以提高 DCT 系数编码的效率。H.261 对符号(游程,幅值)进行编码,并在每个块的结束发送出一个 EOB 字,而 H.263 把 EOB 字结合到 VLC 中。要编码的事件是(最后,游程,幅值),而其中"最后"指示该系数是不是块中最后的一个非零系数。
3. 在块组级以及 MTYPE 和 CBP 编码中降低开销。
4. 支持更多图像格式(表 13.2)。

表 13.2　由 H.261 和 H.263 所支持的图像帧格式

	Sub - QCIF	QCIF	CIF	4 CIF	16 CIF	自定义尺寸
亮度宽度(像素)	128	176	352	704	1 408	< 2 048
亮度高度(像素)	96	144	288	576	1 152	< 1 152
H.261		√	Opt.	Still		
H.263	√	√	Opt.		Opt.	Opt.

"Opt." = 可选;"Still" = 静止图像

除了这些改善外,H.263 还提供了一组定义在标准的附件中的可选特性。

1. 无限定运动矢量(附件 D)。它允许指向图像的外面,在摄像机运动或运动发生在图像边界的情况下可提高编码效率。对指向图像外面的运动矢量的预测信号是由重复图像的边界像素生成的。运动矢量范围扩展到[- 31.5, 31]。
2. 基于语法的算术编码(附件 E)可用于代替可变长(哈夫曼)编码,产生相同的解码图像时,对于 P 帧平均比特率节省 4%,对于 I 帧节省 10%。然而,解码器的计算需求增长了 50% 以上[10]。这将限制实现这个附件的制造商数量[①]。
3. 先进预测模式(附件 F)包括无限定运动矢量模式。先进的预测模式提供了两个附加改善:重叠块运动补偿(OBMC)可用来预测图像的亮度分量,它提高了预测性能并显著减少了块失真(见 9.3.2 节)[59]。在一个 8×8 的亮度预测块中,每个像素是由三个运动矢量计算得到的三个预测值的加权和;这三个运动矢量是:当前宏块的矢量和最靠近当前 8×8 块的两个宏块的矢量。运动估计的加权系数和等价的窗函数已经在图 9.16 及图 9.17中示出了。

   先进的运动预测的第二个改善是一个宏块可选择使用 4 个运动矢量,每个亮度块一个。这样能够较好地模拟真实图像中的运动。然而,这需要编码器来决定在哪个宏块中 4 个运动矢量足以证明编码这些运动矢量所需的额外比特是值得的。这些运动矢量也是预测地来编码 (图 13.7)。
4. PB 图像(附件 G)是把双向预测图像与一般前向预测图像一起编码的一种模式。B 图像在时间上位于 PB 图像的 P 图像之前。与基于逐帧计算的双向预测对比(9.2.4 节,图 9.12),PB 图像使用宏块级的双向预测。在一个 PB 帧中,每个宏块的块数目是 12 而不是 6。在每个宏块内,属于 P 图像的 6 个块先传输,随后是 B 图像的块(图 13.8)。双向预测得自前一个解码的帧和当前宏块的 P 块。如在图 13.8 所看到的,对于在 B 图像和 P 图像之间运动的情况下,这就把后向预测限制于 B 块中那些与当前 P 宏块内的像素对齐的像素 (图 13.8 中的淡灰度区)。对于 B 块中的淡灰度区,通过前向和后

① 这个附件使用了一个次最优的算术编码实现。

向预测结果的平均计算预测。B 块的白色区域中的像素只用前向运动补偿进行预测。一种改进的 PB 帧模式(附件 M)后来被采纳,该模式去除了这种限制,能够得到常规 B 帧的效率 (9.2.4 节)。

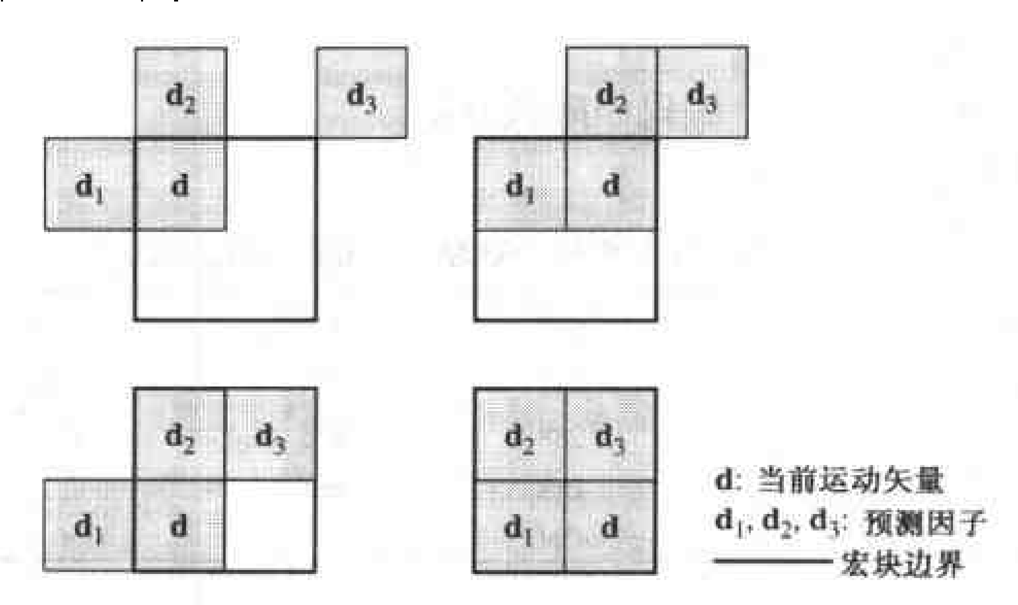


图 13.7 以先进预测模式预测运动矢量,一个宏块中具有 4 个运动矢量。当前运动矢量 $\boldsymbol{d}$ 的分量的预测值是它的预测因子的中值

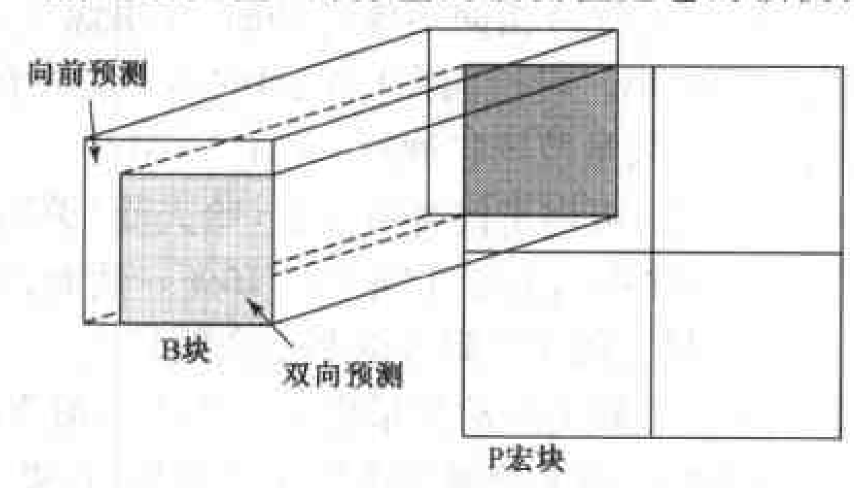


图 13.8 在 PB 帧中,前向预测可用于所有的 B 块;后向预测只用于其后向运动矢量与当前宏块的像素对齐的那些像素

对编码中等幅度运动的图像序列,PB 图像是非常有效的。对于快速运动或复杂运动的场景,或以低帧率编码时,PB 图像的效果不是很好。由于 B 图像的图像质量不影响后续帧的编码,H.263 规定 PB 图像集的 B 图像以低于 P 图像的质量编码,对 P 块使用比相应的 B 块更小的量化器步长。PB 图像增加了编码系统的延迟,因为它们只在后面的 P 帧被摄取和处理后才允许编码器送出 B 帧的比特。这限制了它们对交互式实时应用系统的用途。

由于编码模式的数目较大,编码器的判决比 H.261 中的更复杂。对具有可选项(无限制运动矢量模式和先进预测)的 RD 最优 H.263 编码器与标准开发时所用的测试模型编码器 TMN5 进行了比较[70]。比特率在 20 ~ 70 kbps 之间时,最优编码器的 PSNR 比 TMN5 提高 0.5 ~ 1.2 dB。

2000 年的 H.263 批准了 H.263 的最初版本后,工作继续进行,进一步添加了可选模式。然而,由于存在 15 个以上的可选模式,对于任何制造商来讲实现所有这些可选模式都是有问题的。ITU 认识到了这一点,把推荐模式增加到标准建议中。这里列出以前未提到的最重要的推荐模式。

1. 先进的帧内编码(附件 I):帧内块用左侧的块或上方的块作为预测来进行编码,只要该块也是以帧内模式编码的。这种模式将 I 图像的编码效率提高 10 ~ 15%。

2. 去除块效应滤波器(附件 J):在解码的 8×8 块的边界应用自适应滤波器,以减小块失真。这个滤波器也影响预测图像,它是在编码器和解码器预测环的内部实现的。
3. 补充增强信息(附件 L):这种信息可用来提供由使用 H.263 的应用系统定义的用于外部应用的标志信息。进一步,它可用于通知增强的显示能力,例如帧冻结、缩放或色键(见 10.3 节)。
4. 改进的 PB 帧模式(附件 M):如已提到的,这种模式去掉了附件 G 的加于后向预测的限制。因此,这种模式使常规的双向预测成为可能(9.2.4 节)。

这些工具已经开发出来以提高编码效率。为了使 H.263 视频能在不可靠网络(例如无线网和因特网)上传输,也已开发了一组以差错复原为目的的工具。这些工具包含在附件 H (用 BCH 码的前向纠错)、附件 K(用片结构模式的灵活同步标记插入)、附件 N 和 U(参考图像选择)、附件 O(可分级性)、附件 R(独立分割解码)、附件 V(数据分割和 RVLC)以及附件 W(包头重复)中。在第 14 章描述了这些工具。H.263 的进一步讨论可在参考文献[6]和标准本身[50]中找到。

13.2.3 比较

图 13.9 比较了 H.261 和 H.263 的性能[10]。图中示出了使用和不使用环路滤波器的 H.261 (曲线 3 和 5)。因为 H.261 是为 64 kbps 及以上的数据率设计的,所以我们在这个速率上讨论图 13.9。当没有可选项时,H.263 优于 H.261 近 2 dB(曲线 2 和 3)。如果我们使用先进预测、基于语法的算术编码和 PB 帧这些选项,可以另外获得 1 dB 的增益(曲线 1)。曲线 4 表明,把 H.263 的运动矢量限制于整数像素会降低编码效率达 3 dB。这是由于降低了运动补偿精度和没有低通滤波器所引起的;它的双线性内插引入半像素运动矢量。比较曲线 3 和 5 可看出这种低通滤波器对编码效率的影响。曲线 4 和 5 之间的区别主要是由于变换系数的三维 VLC 编码以及编码 MTYPE 和 CBP 的改进引起的。

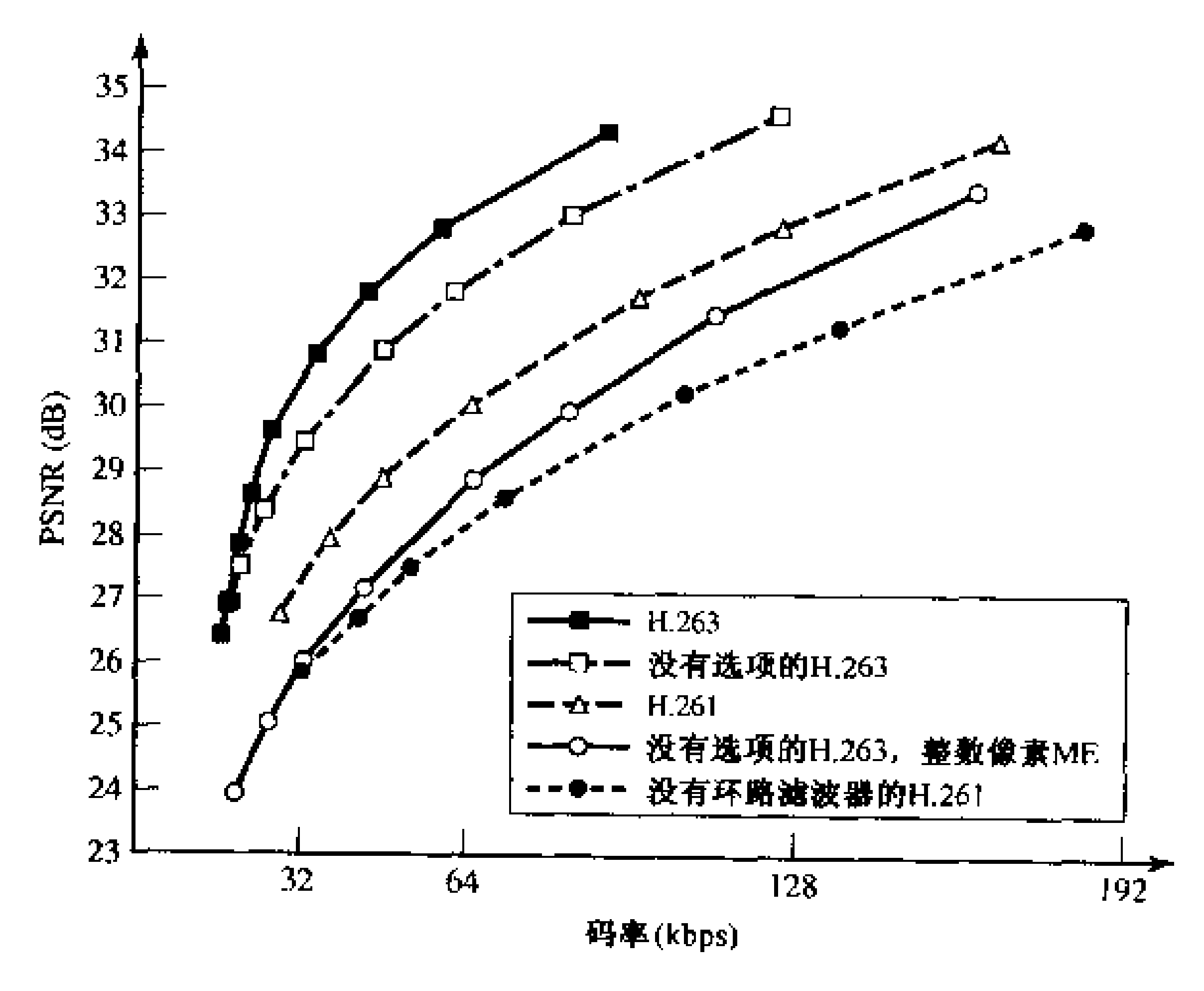


图 13.9 QCIF 格式和 12.5 Hz 帧率下序列"Foreman"的 H.261 和 H.263 的性能

13.3 可视通信系统的标准

为了应用音视频通信,终端必须建立一个公共的通信信道,交换它们的权能,并且在用于交换音视频信息的标准上取得一致。换句话说,我们需要不止一个音频和视频编解码器,以便使音视频通信成为可能。在服务器与客户之间通过网络建立通信是由一个系统标准解决的。ITU-T开发了几个系统标准(包括H.323和H.324)建立不同网络上的双向多媒体通信,开发了几个用于音频通信的音频编码标准以及两个重要的视频编码标准H.261和H.263。表13.3示出了这些系统标准所用的视频、音频、复用和呼叫控制标准的概述[5]。下面,我们简单描述一下最近的标准H.323[51]和H.324[44]的功能。

表13.3 ITU-T多媒体通信标准

网络	系统	视频	音频	复用	控制
PSTN	H.324	H.261/3	G.723.1	H.223	H.245
N-ISDN	H.320	H.261	G.7xx	H.221	H.242
B-ISDN/ATM	H.321	H.261	G.7xx	H.221	Q.2931
	H.310	H.261/2	G.7xx/MPEG	H.222.0/1	H.245
QoS LAN	H.322	H.261/3	G.7xx	H.221	H.242
Non-QoS LAN	H.323	H.261	G.7xx	H.225.0	H.245

PSTN:公共交换电话网;N-ISDN:窄带综合业务数字网(2×64 kbps);B-ISDN:宽带ISDN;ATM:异步传输模式;QoS:保证的服务质量;LAN:局域网;H.262等价于MPEG-2视频[43,17];G.7xx代表G.711、G.722和G728。

13.3.1 H.323多媒体终端

建议H.323[51]提供了在因特网等基于包的网络上运行的多媒体通信系统的技术要求,在这种网络上保证的服务质量经常是不可利用的。

图13.10示出了H.323为在打包网络上传输视频会议所需要的各种协议和标准。H.323呼叫模式可选择地由门管理器允许请求开始(H.225.0 RAS,[48])。然后呼叫信号在通信终端之间建立连接(H.255.0,[48])。接下来为呼叫控制和权能交换建立通信信道(H.245,[49])。最后,用RTP及其相关的控制协议RTCP[65]建立媒体流。一个终端可支持几个音频和视频编解码器,然而,对G.711音频(64 kbps)的支持是强制性的。G.711是目前用于电话呼叫数字传输的公共交换电话网(PSTN)的标准。如果终端要求有视频权能,它必须至少包括一个具有QCIF空间分辨率的H.261视频编解码器[41]。现代H.323视频终端通常使用H.263[50]进行视频通信。

13.3.2 H.324多媒体终端

H.324[44]不同于H.323,当在PSTN上使用V.34[42]调制解调器时,它使相同的网络通信能具有保证的服务质量。H.324可支持多媒体类型的语音、数据以及视频。如果终端支持这些媒体中的一个或多个,那么它就可使用与H.323一样的音视频编解码器。然而,它也支持H.263视频和G.723.1在5.3和6.3 kbps速率的音频[38]。6.3 kbps速率的G.723.1编解码

器的音频质量非常接近常规电话。用 H.245 处理呼叫控制。在 PSTN 上传输这些不同类型的媒体要求按照 H.223 复用标准 [45]进行媒体复用(图 13.11)。复用后的数据经 V.34 调制解调器送到 PSTN,用 V.8 或 V.8bis 程序[53,52]开始和停止传送。如果 H.324 终端使用外部的调制解调器,就要用调制解调器控制 V.25ter[47]。

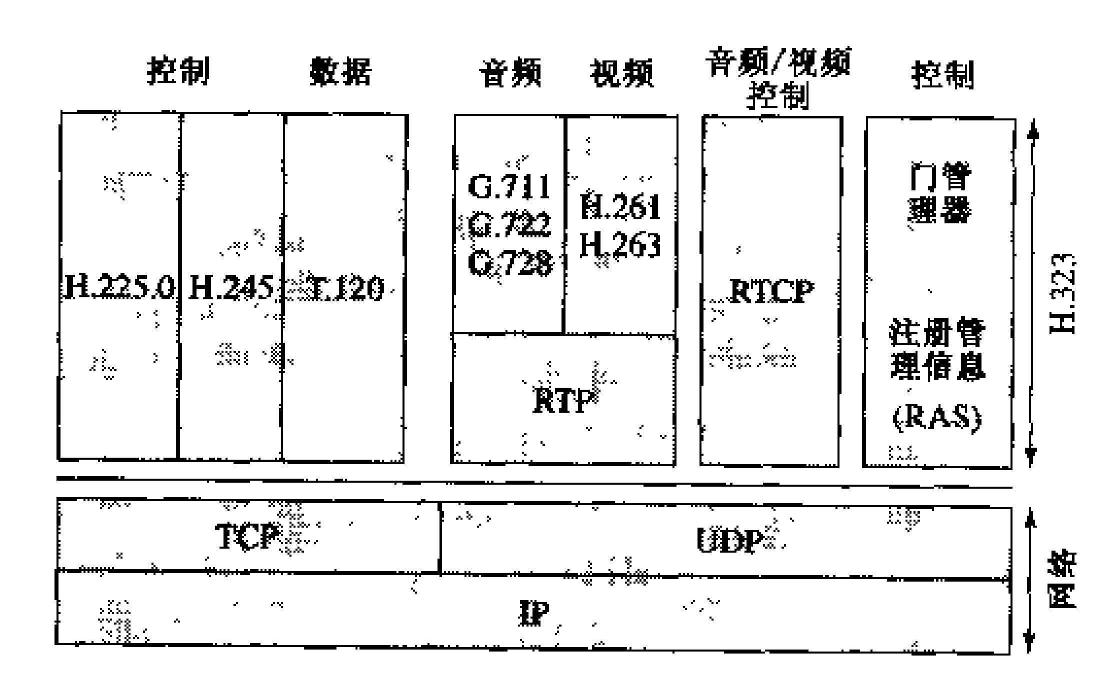


图 13.10 用于 TCP/IP 多媒体通信的 H.323 协议

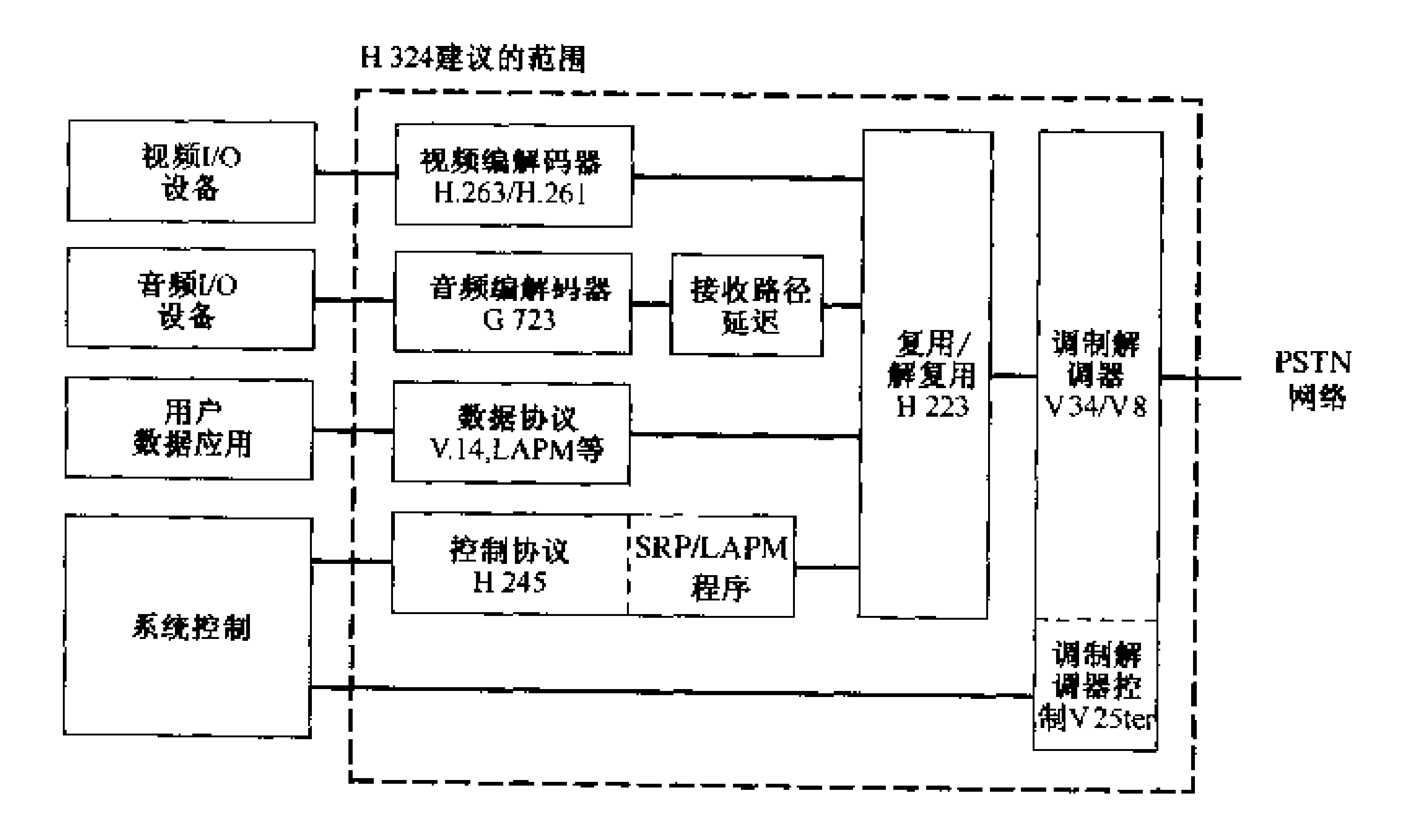


图 13.11 在 PSTN 上的 H.324 多媒体通信系统的方框图

13.4 采用 MPEG-1 的消费视频通信

MPEG 标准是由 Leonardo Chiariglione 任主席的 ISO/IEC JTC1 SC29/WG11 开发的。MPEG-1 是为多媒体应用中所使用的逐行扫描的视频设计的,目标是要在大约 1.2 Mbps(包括音频和视频在内 1.5 Mbps)的比特率下生成接近 VHS 质量的视频。由于预见到大多数的多媒体内容是在 CD-ROM 上发布的,在开发 MPEG-1 时,1.5 Mbps 是 CD-ROM 播放器的访问速率。视频格式是 SIF。最终的标准支持更高的速率和更大的图像尺寸。开发 MPEG-1 时的另一个重要考

虑是实现支持类似录像机(VCR)的基本的交互式功能,如快进、快倒以及每半秒随机地访问存储的比特流[55]。

13.4.1 概述

MPEG-1 标准正式应称为 ISO 11172[16],它由 5 部分组成:系统、视频、音频、一致性以及软件。

MPEG-1 系统提供一个把编码的音频和视频数据组合起来的包结构。它使系统能够把几个音频和视频流复用成一个流,允许各个流同步地重放。这要求所有的流都要以公共的系统时钟(STC)为基准。从这个 STC 中,可导出演播时间标志(PTS);它规定了一个特定的音频或视频帧在终端出现的时间。由于带有 B 帧的视频编码要求重排解码图像,用解码时间标志(DTS)指明什么时候一个特定的图像必须被解码。

MPEG-1 音频是一个不对音频信源性质做任何假设的一般标准。然而,音频编码利用了人类听觉系统的感觉局限以降低不相干性。MPEG-1 音频定义了三层:I,II 和 III。较高的层具有较高的编码效率并要求增加的解码资源。在 20 世纪 90 年代初期的标准化期间,尤其是层 III 由于计算复杂而引起过争议。然而,这个第 III 层 MPEG-1 音频编解码器正是现在每位音乐迷都知道的 MP3。MP3 普及的原因在于声音质量和编码效率,并且最重要的是,在一段时间内,专利的高质量的编码器源代码可以下载到公司的网站上。这开始了音乐产业内的革命(见 13.1.2 节有关成功标准的准则)。

13.4.2 MPEG-1 视频

1989 年 10 月主观测试后开始了 MPEG-1 视频的集中阶段,并导致了发布于 1993 年的标准。因为 1990 年发布了 H.261,所以在 H.261 与 MPEG-1 之间有许多相似之处。图 13.12 示出了 MPEG-1 编码器的概念性方框图。与 H.261(图 13.4)相比较,我们注意到如下的区别:

1. 没有了环路滤波器。由于 MPEG-1 使用半像素精度的运动矢量,因此不需要滤波器(见 13.2.3 节)。运动矢量范围扩大到 ±64 像素。

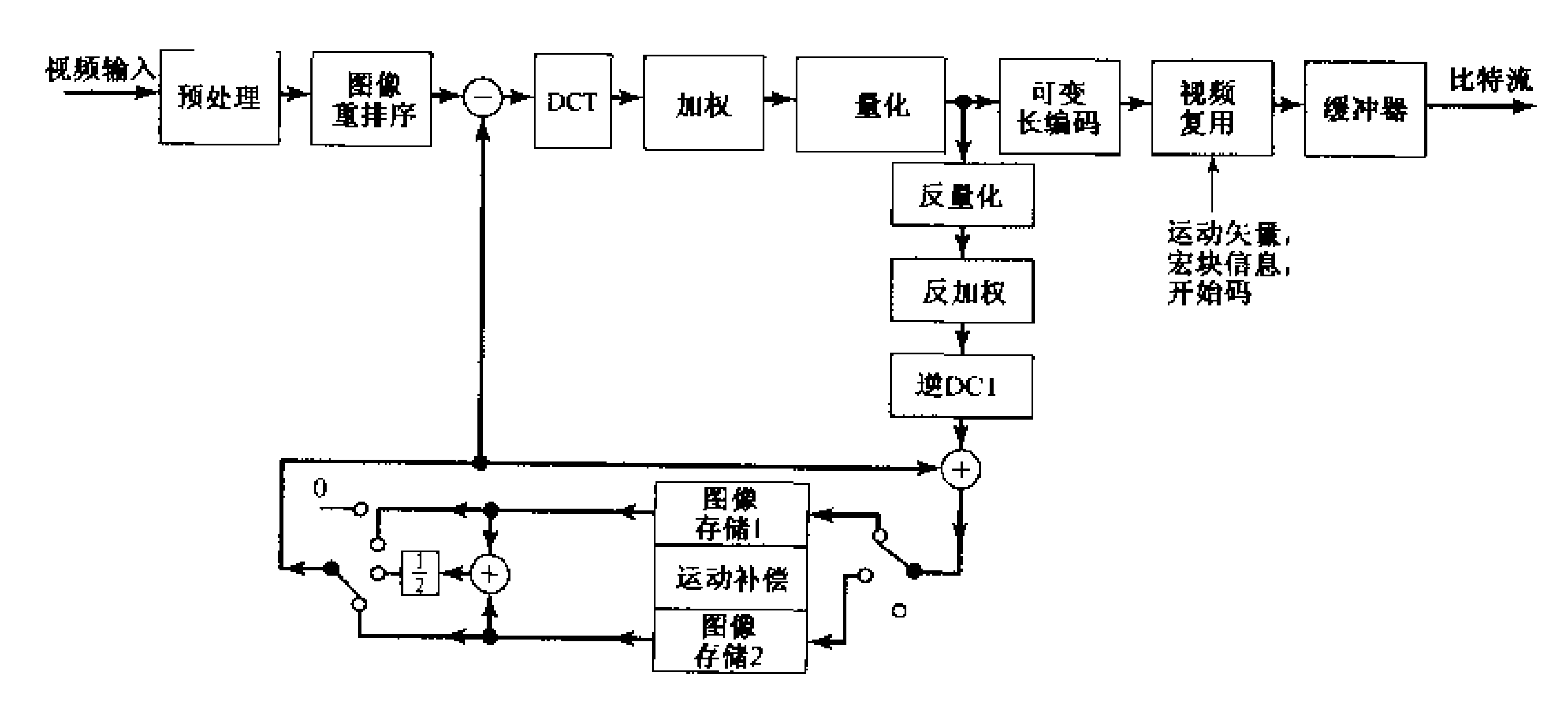


图 13.12 MPEG-1 编码器的方框图

2. MPEG-1 用 I 帧、P 帧和 B 帧。用 B 帧需要一个较复杂的运动估计器和运动补偿单元。B 帧的运动矢量是根据两个参考帧估计的，即前一个 I 帧或 P 帧以及下一个 I 帧或 P 帧。因此，我们可以把两个运动矢量与 B 帧的每一个宏块联系起来。对于运动补偿预测，我们现在需要两个帧存储器用于这两个参考图像。B 帧的预测模式是对每个宏块决定的。此外，编码顺序不同于扫描顺序（见图 9.12），因此，在编码器的输入和解码器中我们需要一个图像重排单元。
3. 对于 I 帧，DCT 系数的量化是适应人类视觉系统的，用一个加权矩阵来除这些系数。图 13.13 示出了该默认表。较大的权值导致系数较粗的量化。可以看到，权值随系数所代表的频率而增加。当在相同的比特率下比较具有和不具有加权矩阵的编码器时，我们注意到，加权矩阵降低了解码图像的 PSNR，但提高了主观质量。

$w_{u,v}$	0	1	2	3	4	5	6	8
0	8	16	19	22	26	27	29	34
1	16	16	22	24	27	29	34	37
2	19	22	26	27	29	34	34	38
3	22	22	26	27	29	34	37	40
4	22	26	27	29	32	35	40	48
5	26	27	29	32	35	40	48	58
6	26	27	29	34	38	46	56	69
7	27	29	35	38	46	56	69	83

图 13.13 用于 MPEG-1 中 I 块量化的默认权值。水平和垂直频率的权值不同

与 H.261 的另一个区别是，一个 I 块的 DC 系数可以由它的左邻块的 DCT 系数预测。这个概念后来扩展到 JPEG[15,40]，H.263 和 MPEG-4 中。

MPEG-1 使用一个图像组（GOP）结构（图 9.12）。每一个 GOP 都以一个 I 帧开始，后跟一定数量的 P 帧和 B 帧。这样能随机地访问视频流，也能实现像 VCR 一样的快进和快倒功能。

因为标准支持更大的比特流特性范围，所以定义了一个特殊的编码参数子集，称为约束参数集（CPS）（表 13.4）。CPS 是一个采样和比特率参数被限定的集，它是为限制解码器计算复杂度、缓冲器尺寸以及存储器带宽而设计的，但它仍能处理具有最大可能的应用范围。用 CPS 实现的解码器只需要 4 Mb 的 DRAM 就能支持 SIF 和 CIF 格式。比特流中的标志位指明它是否为 CPS。

表 13.4 MPEG-1 视频的约束参数集

参数	最大值
像素/行	768 像素
行/图像	576 行
每幅图像中宏块的数目	396 宏块
每秒中宏块的数目	396 × 25 = 330 × 30 = 9 900
输入缓冲器尺寸	327 680 字节
运动矢量分量	± 64 像素
比特率	1.856 Mbps

与模拟消费质量的 VCR 相比，MPEG-1 只用一半扫描行编码视频。然而在 1.8 Mbps 视频比特率下，一个好的编码器所传送的视频质量可能超过由模拟消费 VCR 记录到用过的录像带

上的视频质量。

13.5 采用 MPEG-2 的数字电视

在 MPEG-1 标准化过程即将结束时，已经变得清楚的是，MPEG-1 不能以广播质量有效地压缩隔行数字视频。因此，MPEG 组发出了征集电视广播应用系统的音频和视频数字编码技术建议的通知。性能最优的算法是将 MPEG-1 扩展到处理隔行视频格式。在算法开发的合作实验阶段，保留了大量的与 MPEG-1 的相似性。

MPEG-2 的主要目的是使主要使用 ITU－R BT.601(原来的 CCIR601)4:2:0 格式[35]的隔行图像具有类似 MPEG-1 的功能。目标是产生 4～8 Mbps 码率的电视质量图像和 10～15 Mbps 码率的高质量图像。MPEG-2 要解决 SDTV 或 HDTV 隔行视频的高质量编码问题。这涉及到广泛的应用系统、比特率、分辨率、信号质量和服务问题，包括所有形式的数字存储媒体、电视(包括 HDTV)广播和通信[13]。

MPEG-2 标准[19]由 9 个部分组成：系统、音频、视频、一致性、软件、数字存储媒体——命令与控制(DSM－CC)、先进音频编码器(AAC)、实时接口和 DSM－CC 一致性。在本节中，我们给出关于 MPEG-2 系统、音频、视频以及 MPEG-2 的类概念的一个简单的概述。

13.5.1 系统

MPEG-2 系统必须具有与 MPEG-1 系统的某些兼容性，有差错复原能力，支持 ATM 网络传输，以及在一个流中传送一个以上的电视节目而不要求这些节目的共同时基。MPEG-2 的节目流(PS)前向兼容 MPEG-1 系统流解码器。一个 PS 包含来自单一节目的压缩数据，具有通常在 1～2 Kb 之间直到 64 Kb 的可变长度包的形式。MPEG-2 的传输流(TS)与 MPEG-1 不兼容。TS 提供如有线电视网络或卫星电视所需要的差错复原，它采用 188 字节的包，可能携带具有独立时基的几个节目，易于频道跳转访问。

13.5.2 音频

MPEG-2 音频分为两部分：在标准的第 3 部分中，MPEG 定义了一个支持五通道环绕声的前向和后向兼容的音频格式。其语法设计使 MPEG-1 音频解码器能从 MPEG-2 音频比特流的五个通道向下混合再生出有意义的输出[18]。在第 7 部分中，定义了更有效的多通道音频解码器 MPEG-2 AAC，具有声音效果以及许多其他特性[20]。对于同样的立体声质量，MPEG-2 AAC 比 MPEG-1 层 III 音频所需的比特少 30%。AAC 已被日本广播行业采用；然而，它还没有作为一种格式在因特网上普及，因为没有免费的编码器可利用

13.5.3 视频

MPEG-2 是以电视演播室以及标准清晰度电视和高清晰度电视广播为目标的。因此，它必须有效地支持隔行视频编码，码率应满足应用系统的需要。MPEG-1 与 MPEG-2 之间的主要区别如下：

1. 与 MPEG-1，H.261 和 H.263 相比，4:2:0 格式的色度采样点的位置水平移动 0.5 个像素(图 13.14)。

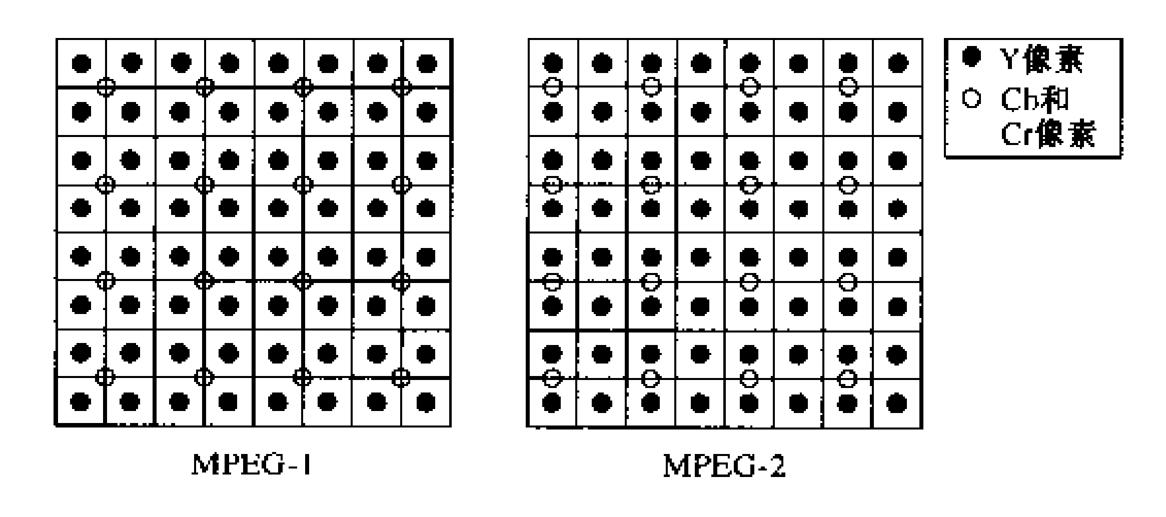


图 13.14　4:2:0 逐行帧中的亮度和色度采样点

2. MPEG-2 能够编码 4:2:0 格式的隔行序列(图 13.15)。

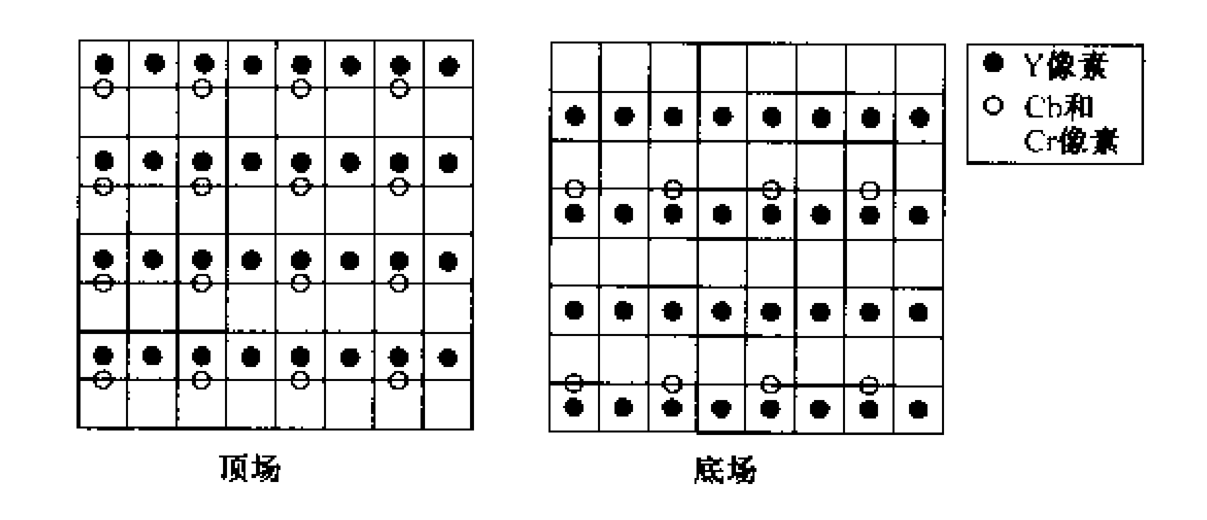


图 13.15　4:2:0 隔行帧中的亮度和色度采样点,其中顶场在时间上居前

3. 由上述结果,MPEG-2 允许 DCT 系数的其他扫描图形和具有 16×8 像素尺寸的块运动补偿。
4. 例如 DCT 的 DC 系数的 10 比特量化、非线性量化和较好的 VLC 表等几个区别也提高了逐行视频序列的编码效率。
5. MPEG-2 支持各式各样的可分级性。空间可分级性使不同的解码器能从同一个比特流中提取出不同图像尺寸的视频。MPEG-2 支持时间可分级性,可把比特流解码成不同帧率的视频序列。此外,SNR 可分级性提供了从同一个比特流中提取出不同幅度分辨率的视频序列的能力。
6. MPEG-2 定义了类和级,它们构成了 MPEG-2 的特性子集及其参数范围,其参数范围在比特流的头中具有标识信号(见 13.5.4 节)。通过这种方法,依从 MPEG-2 的解码器立即可知道它是否能解码这个比特流。
7. MPEG-2 允许高得多的比特率(见 13.5.4 节)。

下面我们将讨论为支持隔行视频和可分级性所引入的扩展。

隔行视频编码　隔行视频是一个顶场和底场交替的序列(见 1.3.1 节)。如果两个场都是顶场或都是底场,它们就具有相同的极性。否则,就称这两个场具有相反的极性。MPEG-2 考虑两种类型的隔行视频图像结构(图 13.16)。帧图像是由隔行图像的顶场和底场的行以隔行顺序组成的。当编码逐行视频时也用这种帧图像结构。场图像保持图像的顶场和底场分离。对每种图像,都可利用 I 图像、P 图像以及 B 图像编码模式。

MPEG-2 添加了新的运动补偿预测模式,所有这些模式都与隔行视频有关:

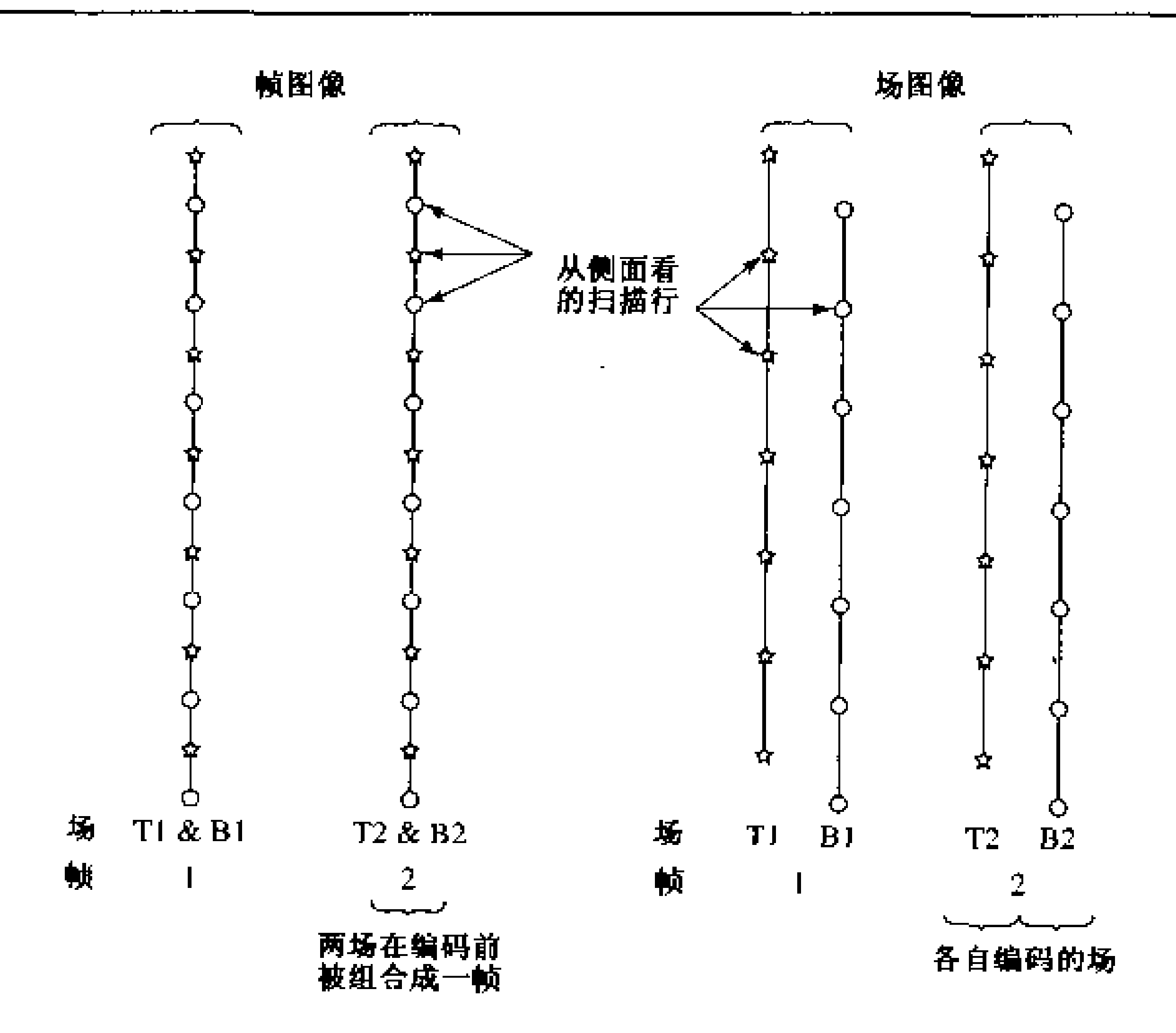


图 13.16 帧和场图像结构(各个场的侧视图):每一帧由一个顶场和一个底场组成;任何一场都可能在时间上居前

1. 场图像的场预测用来预测场图像中的宏块。对于 P 场,预测可以来自两个最近编码的任何一场。对于 B 场,我们使用属于两个参考图像的两个场(图 13.17)。

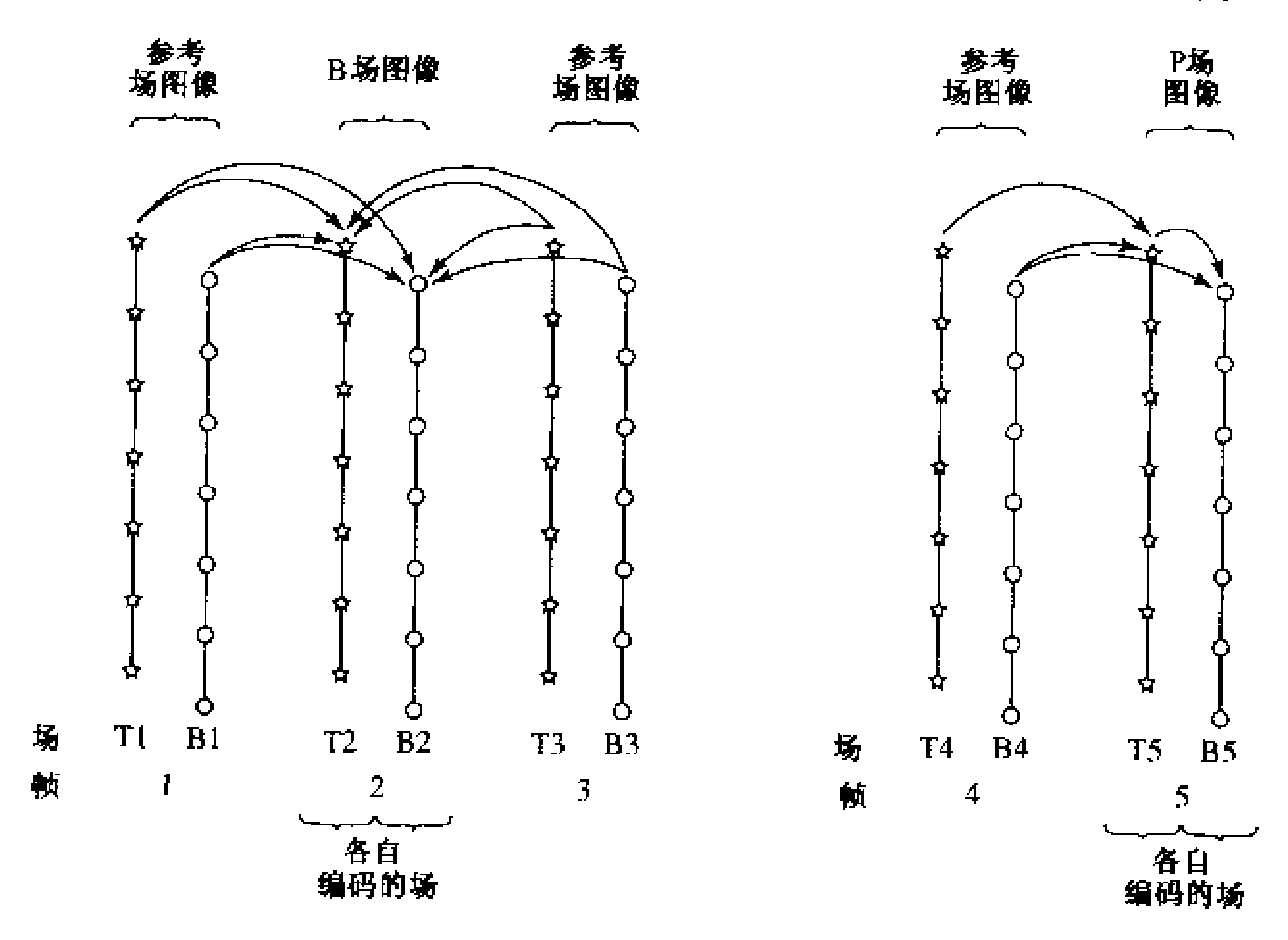


图 13.17 与场图像的场预测有关的每个宏块都位于参考图像的一场之内。图像可以具有不同的极性

2. 帧图像的场预测把该帧的一个宏块分成顶场像素和底场像素,产生两个 16×8 的场块(图 13.18)。每一个场块是独立于另一个且用类似于刚才介绍的场图像方法进行预测

的。这种预测方法特别适用于快速运动。

图 13.18 帧图像的场预测:要预测的宏块分成顶场像素和底场像素。每个 16×8 的场块用它自己的运动矢量(P帧)或两个运动矢量(B帧)分别进行预测

3. P 图像的双基为每个宏块传输一个运动矢量;这个运动矢量可用于由前面的 P 图像或 I 图像来预测场和帧图像。目标宏块被表示为两个场块。编码器为每个场块计算两个预测值并平均它们。每个场块的第一个预测值是通过运动补偿计算的,在计算中使用所传输的运动矢量和具有相同极性的参考场。每个场块的第二预测值是用修正运动矢量和具有不同极性的参考场计算的。修正运动矢量是在假设线性运动的条件下计算的。考虑到同极性场之间的时间距离,所传输的运动矢量被缩放以反映反极性场之间的时间距离。然后与一个所传输的差分运动矢量(DMV)相加,产生修正的运动矢量。对于隔行视频,这种 P 图像的双基预测模式具有与使用 B 图像一样的效率,但不增加 B 图像的延迟。
4. 场图像的 16×8 MC 对应于帧图像的场预测。在一个宏块内,属于不同场的像素具有它们自己的用于运动补偿的运动矢量;也就是说,为 P 图像传输两个运动矢量,为 B 图像传输四个运动矢量。

如此多的预测选择显然使最优编码器的设计很具有挑战性。

在隔行视频中,一个宏块中的相邻行来自于不同的场,因此当场景包含具有垂直分量的运动时,就会降低行之间的垂直相关性。MPEG-2 提供了两种新的编码模式来增加预测误差编码的效率。

1. 场 DCT 把一个宏块的像素重新组织成顶场的两个块和底场的两个块(图 13.18)。这就增加了运动情况下块内的相关性,从而增加了编码效率。
2. MPEG-2 提供了一种交替扫描,编码器可以基于图像更替选择这种扫描。这种扫描把具有高垂直频率的系数放得比"Z"形扫描靠前。图 13.19 比较了这种新的扫描与传统的"Z"形扫描。

MPEG-2 的可分级性 至此所描述的 MPEG-2 的功能性是用 MPEG-2 的不可分级语法实现的,它是 MPEG-1 的一个超集。可分级语法用层来构造比特流。基本层可以使用不可分级语法,因此它可以被一个不理解可分级语法的 MPEG-2 终端解码。基本的 MPEG-2 可分级性工具是数据分割,SNR 可分级性,空间可分级和时间可分级性(见 11.1 节)。也支持这些基本可分级性工具的组合。

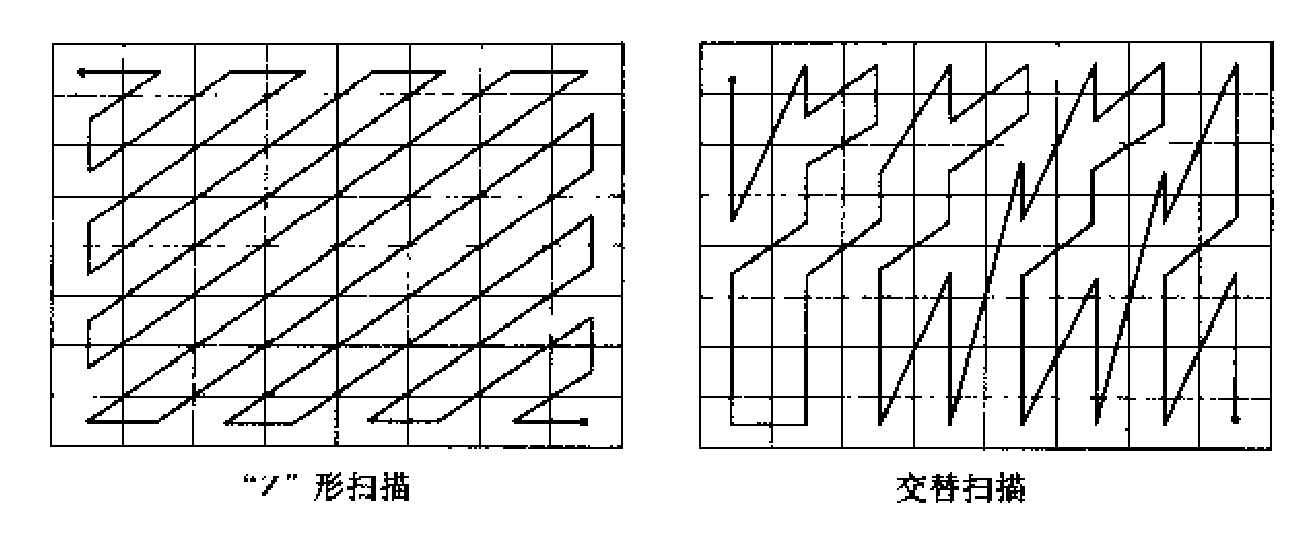


图 13.19 从 H.261、H.263 和 MPEG-1 中已经知道的"Z"形扫描，在 MPEG-2 中被交替扫描加以扩充，便于隔行块的编码，这些块在水平方向比在垂直方上有更强的相关性

当使用可分级编解码器时，在只解码基本层的解码器中可能出现漂移。当用于运动补偿的参考图像在编码器与在基本层解码器中不同时，就会产生漂移。如果编码器在计算基本层的参考图像时使用了增强层的信息就会发生这种情况。在每个 I 帧漂移被自动设置为零。如果编码器在编码基本层时不用增强层的任何信息，漂移就不会出现在可分级编解码器中。此外，解码器解码除基本层外的其他层可以不把较高层的数据引入到较低层的解码中。

数据分割：数据分割把视频比特流分为两层或更多的层。由编码器决定哪些语法元素放在高优先权的基本层，哪些放在低优先权的增强层。一般，在低优先权的增强层中传输高频 DCT 系数，而所有的头、辅助信息、运动矢量和前几个 DCT 系数在高优先权的基本层中传输。如果可利用两个传输信道，则数据分割是合适的。由于数据分割，只当解码器备有高层比特流丢失隐藏器时，解码器才能解码基本层。这种隐藏器是简单的，只需把丢失的增强层中的高阶 DCT 系数置为零。图 13.20 示出了编码器和解码器的一个高级示意图。数据分割功能可以独立于编码器和解码器实现。数据分割不引起任何值得注意的开销。然而，在容易发生错误的环境中它的性能可能比其他的可分级性方法差[13]。显然，如果我们只解码基本层，我们将会遇到漂移问题。

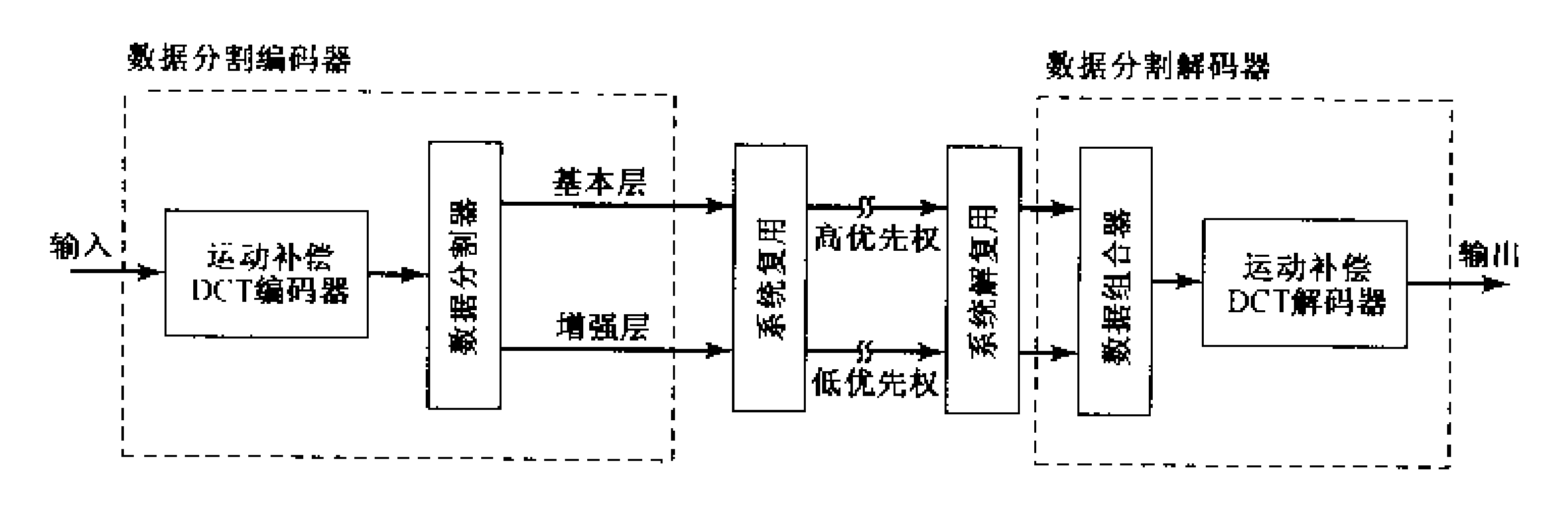


图 13.20 适用于支持两级服务质量 ATM 网络的数据分割编解码器

SNR 可分级性：SNR 可分级性是一种频域方法，在这种方法中，所有的层用相同的空间分辨率编码，但通过不同的宏块量化步长实现不同的图像质量。低层提供基本的视频质量，而增强层携带的信息当加入到低层时能产生输入视频的较高质量的再生。图 13.21 示出了一个 SNR 可分级编码器，它包括一个不可分级的基本编码器。经过变换和量化后，基本编码器把 DCT 系数送入 SNR 增强编码器。增强编码器对基本编码器的量化误差进行再量化，并把送到

SNR 增强解码器的系数反馈回基本编码器，基本编码器把它们与它的反量化的系数相加并送到编码器的反馈环中。由于增强层在解码器中的反馈，对于任何只解码基本层的解码器都会出现漂移。

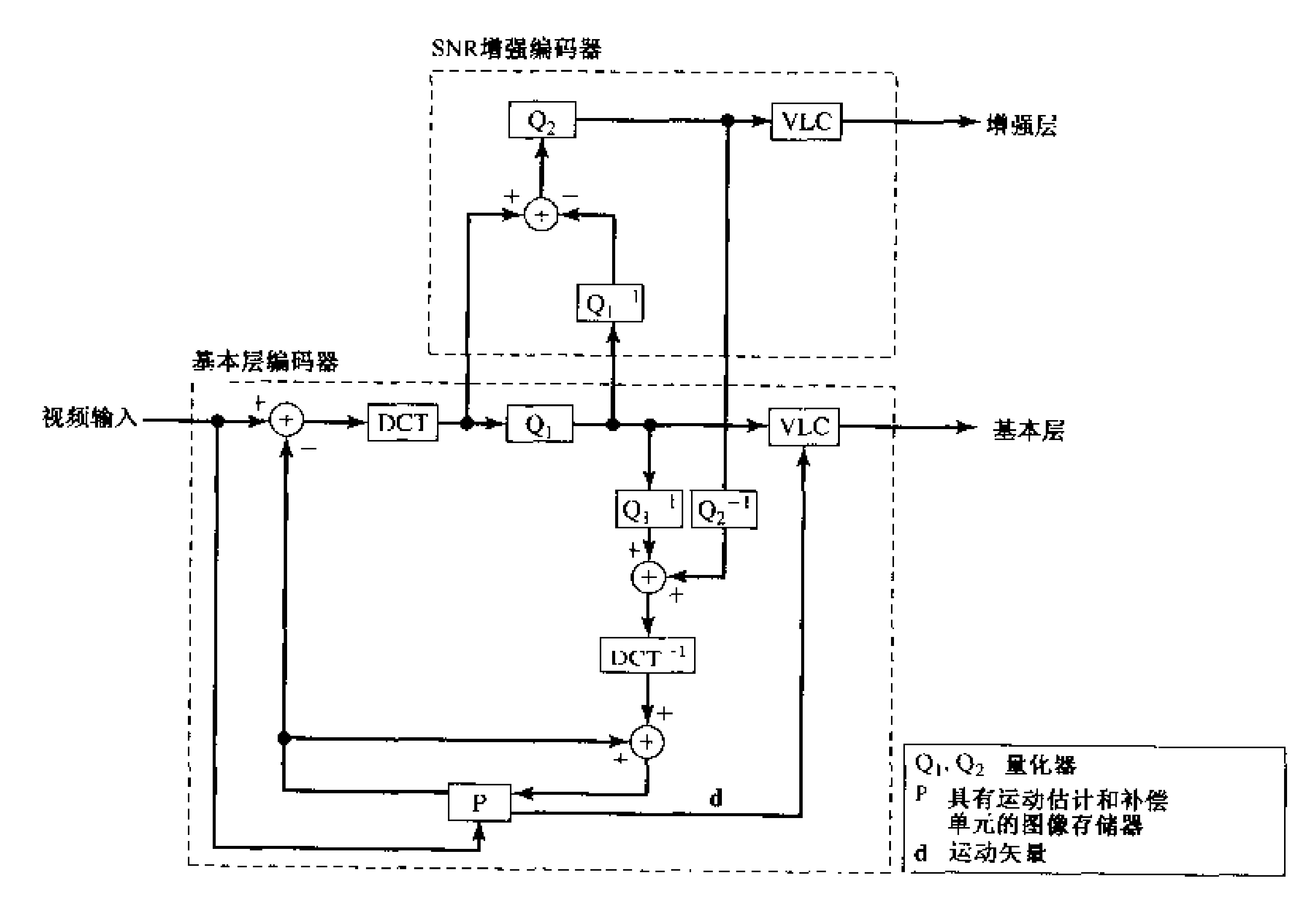


图 13.21 SNR 可分级编码器详图。如果去掉增强编码器，这种编码器就默认为一个标准编码器

在 4 ~ 9 Mbps 的总比特率下，基本层和增强层组合的图像质量比用不可分级编码得到的图像质量低 0.5 ~ 1.1 dB。显然，就基本层的图像质量而言，SNR 可分级性优于数据分割[61,13]。

空间可分级性：在 MPEG-2 中，空间可分级性是通过联合发送端的两个完整编码器和接收端的两个完整解码器实现的。基本层是以低空间分辨率编码的，使用运动补偿 DCT 编码器，例如 H.261，MPEG-1 或 MPEG-2（图 13.22）。在基本编码器反馈环的帧存储器中的图像能够被空间增强编码器所利用。增强编码器也是一个运动补偿 DCT 编码器，它以高分辨率编码输入序列。它使用低层的向上采样的输入以加强它的时间预测。增强层编码器中的预测图像是增强编码器的时间预测图像与基本编码器的空间预测图像的加权和。权重可以是宏块级自适应的。这种编码器不存在漂移问题，因为编码器和解码器都不把增强层的信息引入基本层。在总比特率 4 Mbps 下，基本层和增强层的联合图像质量比用不可分级编码得到的图像质量低 0.75 ~ 1.5 dB[13]。

与同时联播（即传送两个独立的比特流，一个具有基本层分辨率，一个具有增强层分辨率）相比较，空间可分级性效率提高 0.5 ~ 1.25 dB[13,62]。空间可分级性适用于需要视频标准协同工作的应用系统，而且编码效率的提高（与同时联播相比）能够弥补编码器和解码器复杂性所带来的额外开销。

时间可分级性：在时间可分级性中，基本层是以较低的帧率用不可分级编解码器编码的，而中

间的帧可以被编码在另一个比特流中,用第一个比特流的重建作为预测[63]。MPEG-2 规定只能用两帧进行增强层图像的预测。图 13.23 和图 13.24 示出了两种典型的结构。当我们在心里去除图 13.23 中增强层与基本层图像的界限时,我们注意到所得到的图像序列和预测排列与不可分级编码器相似;如果基本层只用 I 帧和 P 帧,就等价于不可分级编码器。相应地,时间可分级的图像质量只比不可分级编码器低 0.2~0.3 dB[13]。在图 13.25 中,我们看到增强层和基本层的编码器是两个完整的编解码器,二者的工作速率都是视频序列速率的一半。因此,时间可分级性的计算复杂度与工作在输入序列全频率上的不可分级编码器类似;不存在漂移问题。

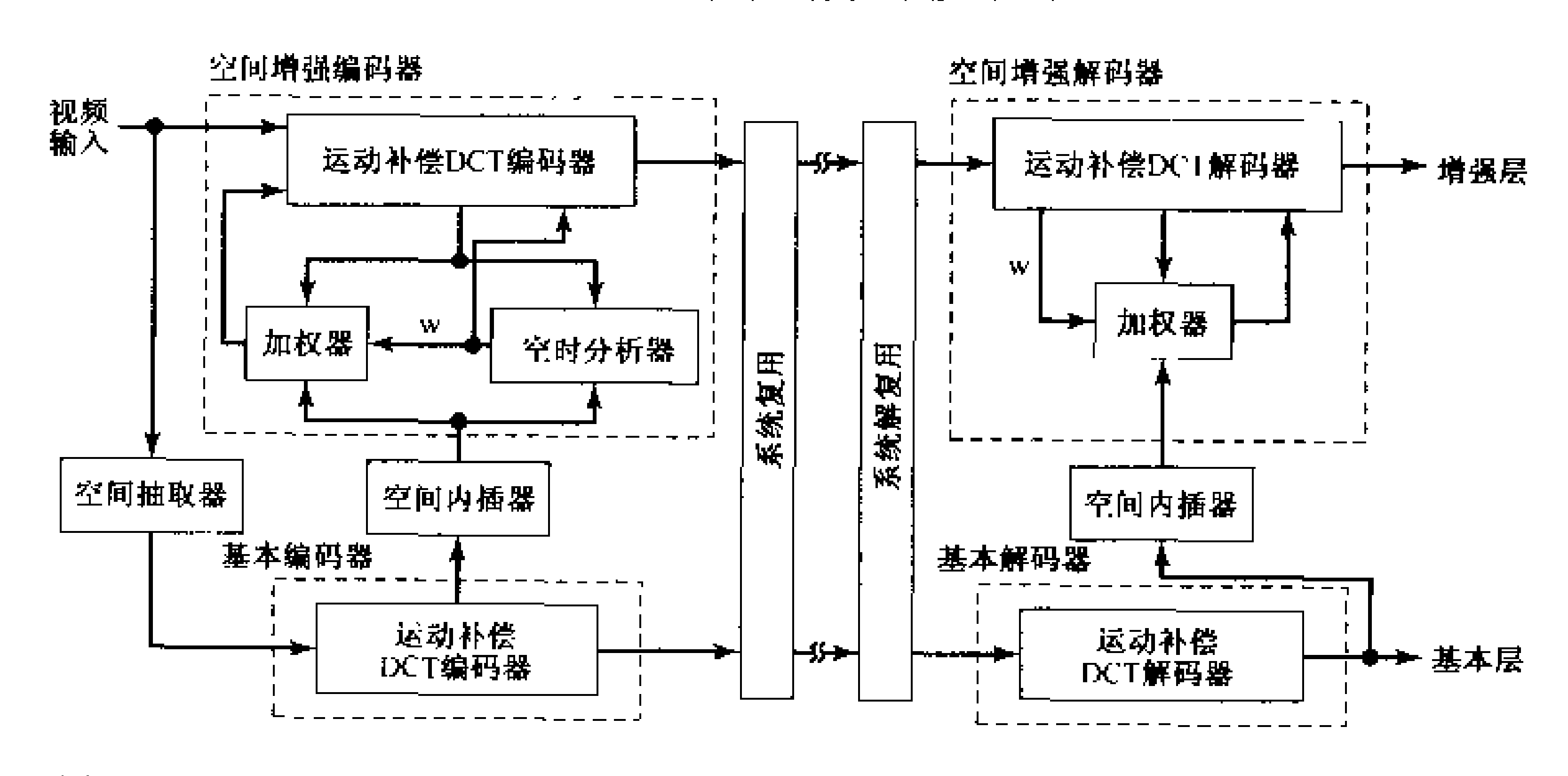


图 13.22 一个空间可分级编码器,它由两个用空间内插滤波器连接起来的完整的编码器组成

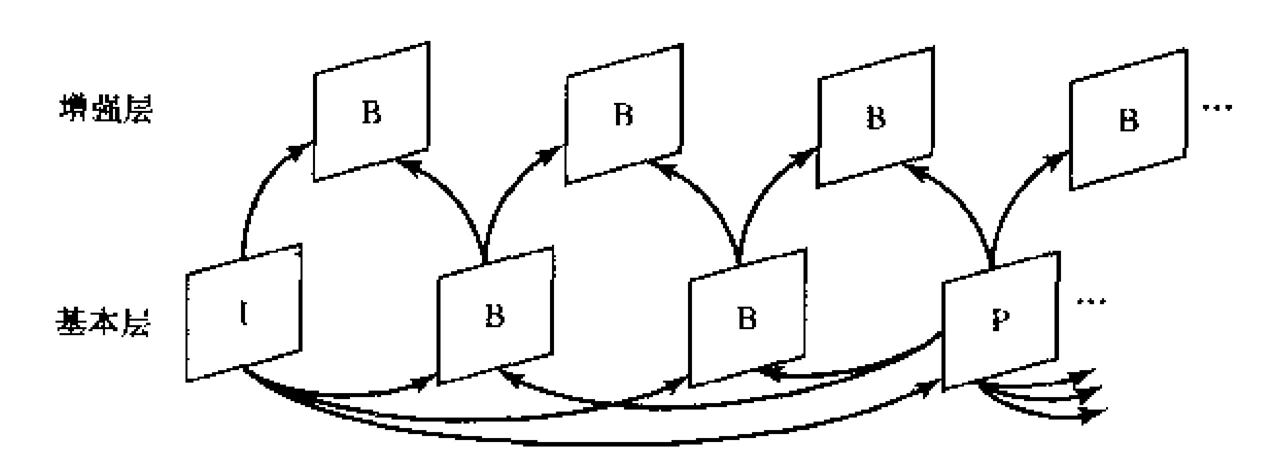


图 13.23 只用基本层预测增强层图像的时间可分级性的结构。显然,增强层中的差错不会随时间扩散

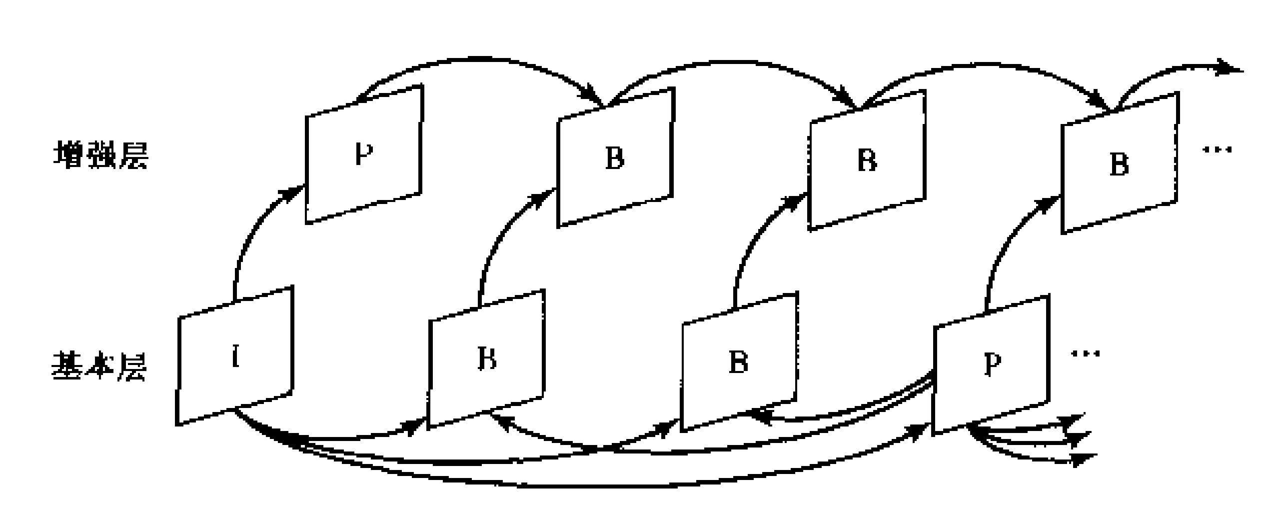


图 13.24 增强层可用基本层和增强层预测的时间可分级性的结构。这种排列特别适用于体视视频的编码

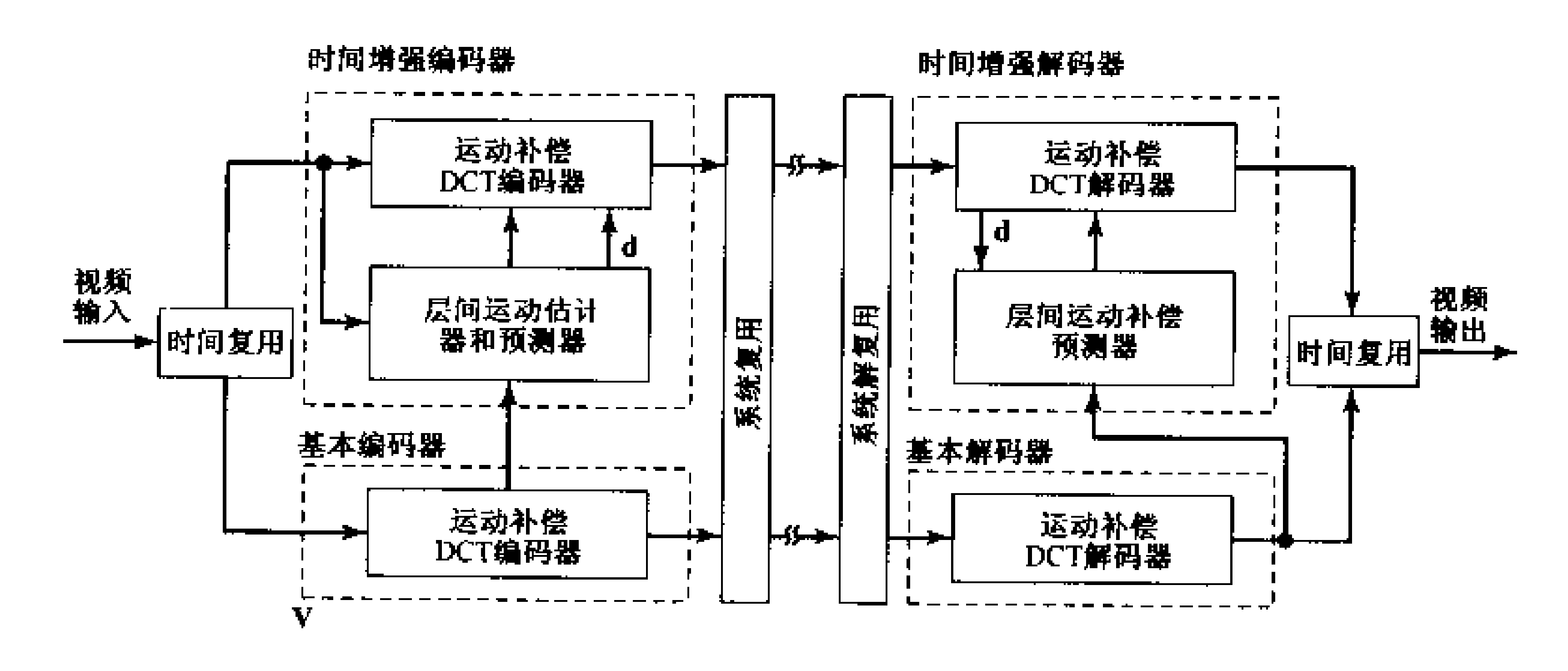


图 13.25　一个由两个完整的编码器组成的时间可分级编码器，增强编码器用基本层的视频作为预测的附加参考。时间解复用把图像交替地送给基本编码器和增强编码器

时间可分级性是一种把视频分配给具有不同计算能力的终端（例如移动终端和台式 PC 机）的有效方法。另一种应用是体视视频传输；其中左右信道被分别作为增强层和基本层传送。这已在 12.5.1 节中讨论过了。

13.5.4　类

全部的 MPEG-2 语法涉及范围广泛的特性和参数。MPEG-2 扩展了 MPEG-1 的受约束参数集的概念（表 13.4），定义了类来描述解码比特流所需要的工具，以及定义了级来描述这些工具的参数范围。MPEG-2 起初为视频定义了 5 个类，每一类以分等级的形式添加新的工具。后来，又添加了两个不适合分层方案的类。

1. 简单类支持 I 帧和 P 帧、4:2:0 格式以及不可分级性。目前它未在市场上使用。
2. 主类增加了对 B 帧的支持。主级中的主类（MP@ML）用于电视广播，这个类得到了广泛应用。
3. SNR 类除了主类的功能外还支持 SNR 可分级性。目前它未在市场上使用。
4. 空间类支持 SNR 类的功能并增加了空间可分级性，目前它未在市场上使用。
5. 高类支持空间类的功能并增加了对 4:2:2 格式的支持。这个类过于复杂而难于应用。
6. 4:2:2 类支持演播室后期制作以及用于存储和发布的高质量视频。它基本上是把主类扩展到更高的比特率和质量。在一个帧组中推荐的帧顺序是 IBIBIBIBI……，使用这个类的设备用于数字演播室中。
7. 多视类能够并行地传输几个视频流，因此能够立体显示。这种功能是用时间可分级性实现的，因此使主类解码器能接收其中的一个视频流。有样机。

MPEG-2 为每个类定义了级。级实际上定义了视频帧的大小、帧率和图像类型，为解码器的处理能力提供了一个上限。表 13.5 示出了为大多数类所定义的级。在表 13.5 中只有两栏（MP@ML 和 4:2:2@ML）在市场上得到应用的事实有力地表明标准化是一个一致同意的过程——MPEG 必须容纳许多个体的愿望，在不为主要的应用系统（即电视生产和广播）增添麻烦的前提下，使所需要的专利技术形成一个类。

表 13.5 在 MPEG-2 中的类和级

			类						
			简单 (I,P) (4:2:0)	主 (I,P,B) (4:2:0)	SNR (I,P,B) (4:2:0)	空间 (I,P,B) (4:2:0)	高 (I,P,B) (4:2:0; 4:2:2)	多视 (I,P,B) (4:2:0)	4:2:2 (I,P,B) (4:2:0; 4:2:2)
级	低	像素/行			352	352		352	
		行/帧			288	288		288	
		fps			30	30		30	
		Mpbs			4	4		8	
	主	像素/行	720	720	720		720	720	720
		行/帧	576	576	576		576	576	512/608
		fps	30	30	30		30	30	30
		Mpbs	15	15	15		20	25	50
	高 1 440	像素/行	1 440		1 440	1 440	1 440		
		行/帧	1 152		1 152	1 152	1 152		
		fps	60		60	60	60		
		Mpbs	60		60	80	100		
	高	像素/行	1 920			1 920	1 920	1 920	
		行/帧	1 152			1 152	1 152	1 152	
		fps	60			60	60	60	
		Mpbs	80			60	130	300	

I,P,B:可允许的图像类型。在可分级比特流情况下,最大比特率包括所有的层。

13.6 采用 MPEG-4 的音视频对象编码

设计 MPEG-4 标准是为了在支持传统应用的同时,满足新一代高度交互性多媒体应用系统的需求。这些多媒体应用系统除高效编码外,还要求先进的功能,例如各个对象的交互性、内容的可分级性和高度的差错复原性。MPEG-4 提供自然的和合成的音频、视频以及图形的基于对象的编码工具。类似于在它以前的标准,MPEG-4 标准由若干部分组成,主要部分是系统、视频和音频。MPEG-4 的音频和视频部分分别包括自然和合成的视频和音频的编码。

13.6.1 系统

MPCE-4 系统能够把音视频对象及其组合复用成一个场景。图 13.26 示出了一个在接收机里组合的、并在显示器和扬声器演播的场景。可提供鼠标和键盘供用户输入。如果我们忽略用户输入,演播就像在常规的 MPEG-1 或 MPEG-2 终端上一样。然而,音视频对象是在接收终端被组合成场景的,而本章中所讨论的所有其他标准要求场景合成在编码之前完成。图 13.26 中的场景是在本地的三维坐标系统中合成的。它包括一个二维背景、一个场景中的在屏幕上播放的视频、一个作为二维子图形(sprite)对象编码的伴随有声音的节目主持人以及桌子和地球仪这样的三维物体。(子图形对象是一个用仿射映射来映射到图像平面上的视频物

体。)MPEG-4提供了与这样的场景相互作用的工具,从而使用户具有交互能力。显然,这种基于对象的内容描述,为创建交互性内容和面向观看者的在语言、文本、广告、理念等方面用户化的演播带来了极大的灵活性。

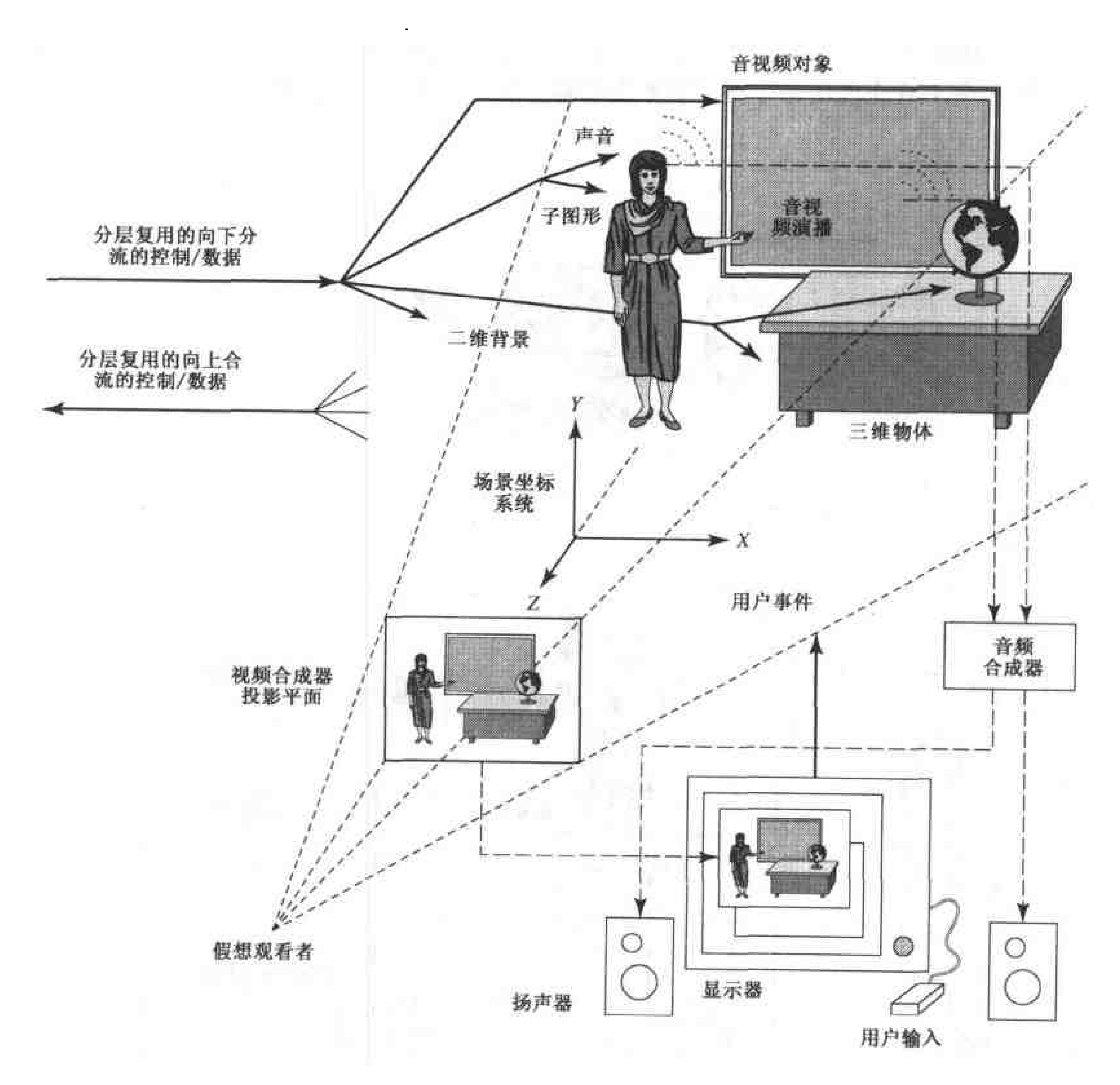


图 13.26 音视频对象在MPEG-4演播的接收机内被组合成一个场景

图13.27示出了MPEG-4终端的各种功能组件[1]:

1. 媒体或压缩层:这是执行媒体(如音频、视频、图形以及其他适当的媒体)解码的系统组件。媒体是通过基本流接口从同步层抽取出来的。专用的MPEG-4媒体包括一个二进制格式场景(BIFS),用以指定场景合成和图形内容。另一个专用的MPEG-4媒体类型是对象描述符(OD)。OD包含指向基本流的指针,类似于URL。基本流被用来传送单独的MPEG-4媒体。OD也包含附加信息,例如服务质量参数。这一层是媒体有意识,而传送无意识的,即它不考虑传输[67]。

2. 同步或基本流层:这个系统组件负责各个压缩媒体的同步和缓冲。它接收来自传输层的同步层(SL)包,根据基本流的时间标志对它们解包,并把它们转发到压缩层。一个完整的 MPEG-4 节目以不同的基本流传送每一个媒体类型。如果涉及到可分级性,某些媒体可以在几个基本流中传输。这一层是媒体无意识且传送无意识的;它通过传送多媒体集成框架(DMIF)应用接口(DAI)与传送层对话。除了通常的通话建立和流控制功能外,DAI 也能使每个流的服务质量需求得到设置。DAI 是独立于网络的[14]。

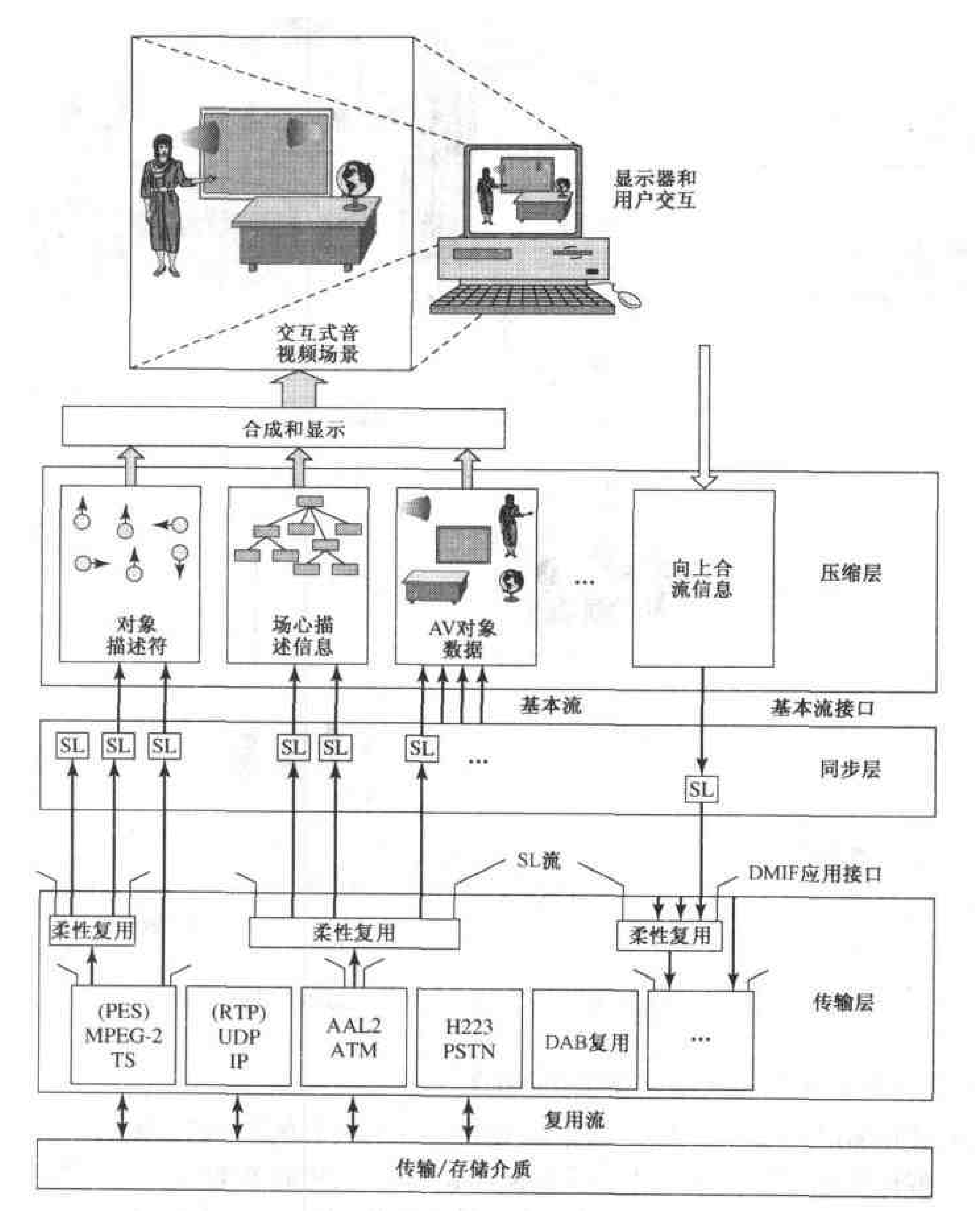


图 13.27 一个包括传输层、同步层和压缩层的 MPEG-4 终端。MPEG-4 不对实际的合成和显示标准化

3. 传输层:传输层是媒体无意识和传输有意识的。MPEG-4 没有定义任何专用的传输层。MPEG-4 媒体可以在现有的传输层(例如 RTP、MPEG-2 传输流、H.223 或 ATM 等)上,用

如参考文献[31,2]中说明的 DAI 传输。

MPEG-4 的二进制格式场景(BIFS)　BIFS 场景模型是虚拟现实模拟语言(VRML) 的一个超集[21,11]。VRML 允许用虚拟摄像机描述和显示一个合成场景中的合成三维物体。MPEG-4在三各方面扩展了 VRML。

1. 定义了二维场景描述,以便把二维音视频对象放到屏幕上。如果所编码的媒体只是不要求三维显示开销的视频流,这是重要的。二维和三维场景可以混合起来。图 13.28 示出了一个在屏幕上放有几个二维对象的场景图。对象的位置用变换节点定义。某些对象是需要三维着色的三维物体。着色后,这些物体被作为二维对象并放入二维场景中。
2. BIFS 用它的基于算术编码器的新压缩工具能进行场景和图形物体的描述和动画。
3. MPEG-4 认识到人脸和身体的特殊重要性。它引入了特殊工具,能够很有效地实现虚拟人的描述和动画。

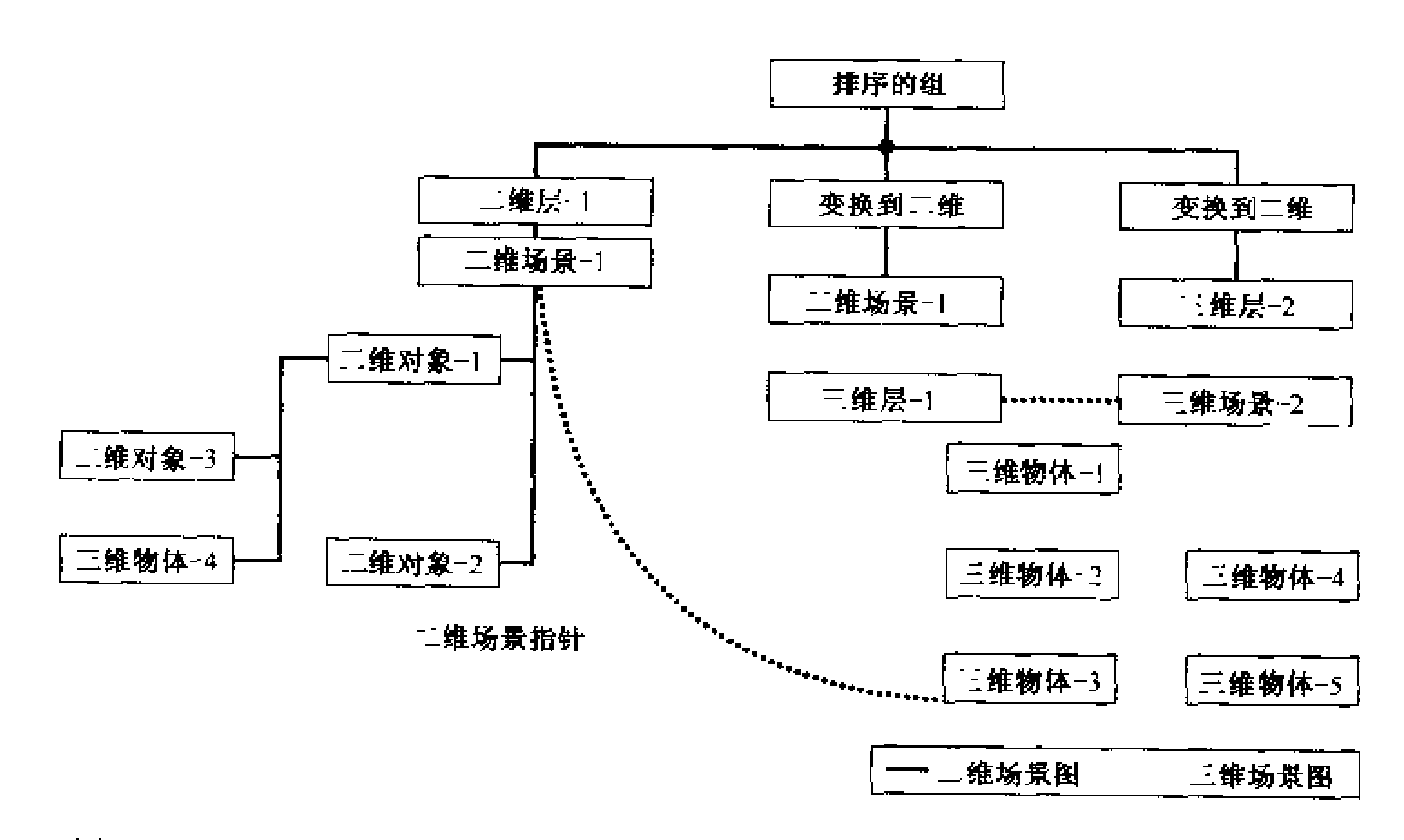


图 13.28　具有二维和三维组件的场景图。二维场景图只需要简单地用“变换到二维”节点把二维对象放到图像中。三维物体被着色,然后把它放到三维层节点定义的屏幕上。对象之间的相互作用可以用由一个节点到另一个节点的指针定义

13.6.2　音频

MEPG-4 音频定义的工具[30,3]可以合并到各种音频编码算法。由于没有找到单一的编码样式覆盖从极低比特率语音信号编码到高质量多通道音频编码的整个范围,因此为宽范围的预期应用定义了一系列不同的算法以实现最优的编码效率(图 13.29)。可分级音频编码器可分成几个组件。

- 在其最低码率下,使用 MPEG-4 TTS 接口(TTSI) 支持文本到语音(TTS)合成器。
- 低码率语音编码(3.1 kHz 带宽)是基于 2 ~ 4 kbps 的谐波矢量激励编码(HVXC)编码器。

- 电话语音(8 kHz 带宽)和宽带语音(16 kHz 带宽)用码激励线性预测(CELP)编码器以3 850 ~ 23 800 bps 的码率进行编码。这个 CELP 编码器可生成具有五层的可分级比特流。
- 用更高效的改进的 MPEG-2 AAC 编码器对每通道 16 kbps 到 64 kbps 以上的一般音频进行编码。可达到明晰的音频质量。

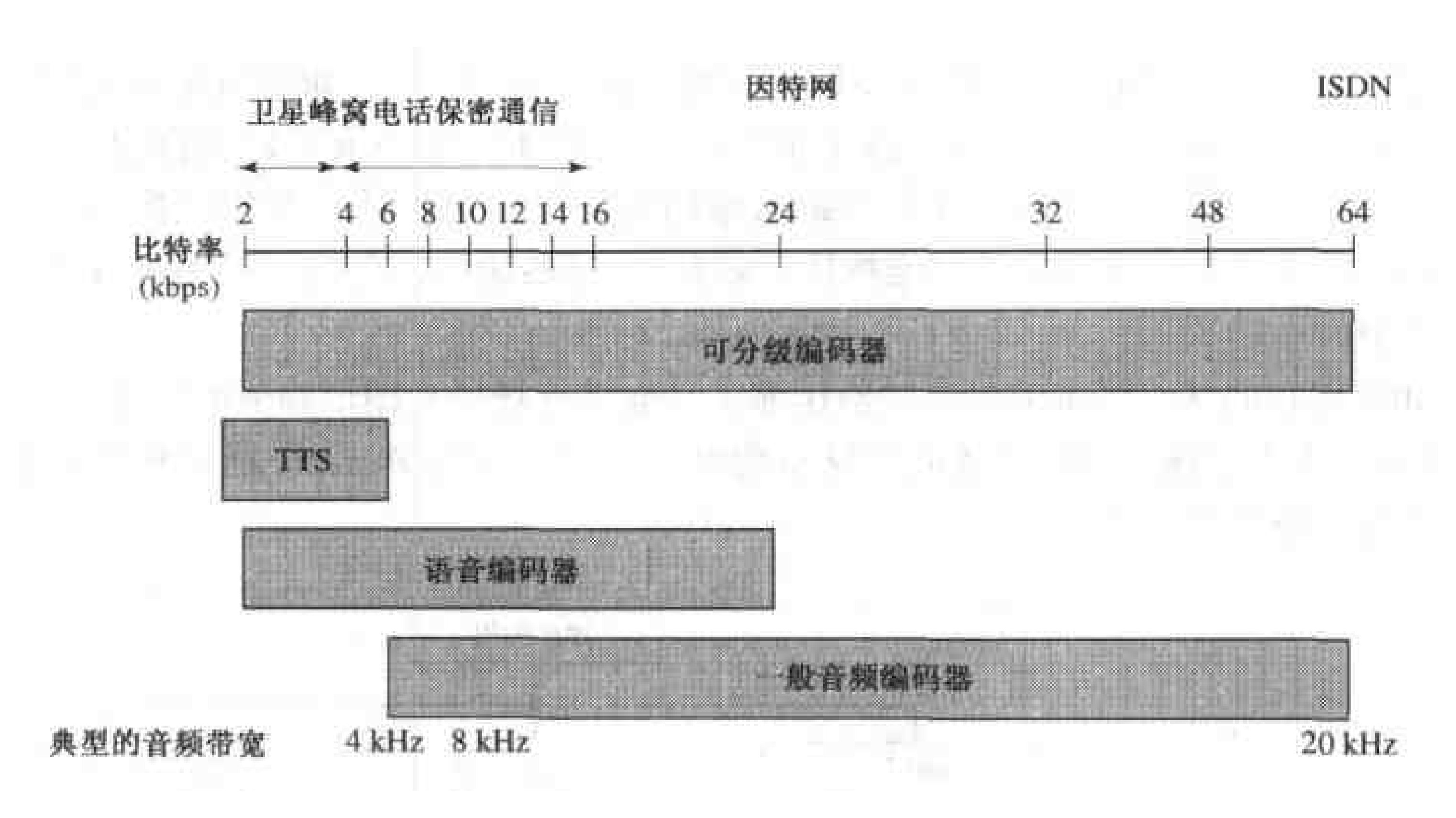


图 13.29 MPEG-4 音频支持 2 ~ 64 kbps/信道的多通道音频编码范围内的语音和音频编码

除了音频编码外,MPEG-4 用结构化音频工具集定义了接收机的音乐合成;它为统一算法音乐合成、实现可分级性和音频对象的表示提供了一个单一的标准。

13.6.3 基本视频编码

许多 MPEG-4 的功能不仅需要访问整个图像序列,而且需要访问整个对象——进一步说,不仅需要访问各个图像,而且需要访问图像内这些对象的时间样本。视频对象的时间样本可以被认为是在图像内出现的任意形状物体的瞬像。与图像相同的是,一个对象也被规定为一个访问单元;与图像不同的是希望它具有语义上的意义。MPEG-4 通过用运动、纹理和形状对物体独立地编码,使与视频对象进行基于内容的交互成为可能。在解码器中,不同的对象综合成一个场景并加以显示。为了实现这个功能,需要开发较高级的句法结构。一个场景是由几个视频对象(VO)组成的。VO 具有 3 个维(二维加时间)。一个 VO 可由几个视频对象层(VOL)组成。每个 VOL(二维加时间)表示 VO 的各种实例。VOL 可以表示可分级比特流的不同层或 VO 的不同部分。VOL 的一个时间瞬间称为一个视频对象平面(VOP)。一个 VOP 是一个矩形视频帧或它的一部分。可由其纹理变化(一组亮度和色度值)和其形状完全描述。视频编码器把运动、纹理和形状编码工具用于 VOP,使用的 I,P 和 B 模式类似于 MPEG-2 的模式。为了编辑和随机访问,相继的 VOP 可组成一个视频对象平面组(GVOP)。视频会话是最高的语法结构,可由几个 VO 组成。

图 13.30 中的例子示出了由两个 VOL 组成的一个 VO。VOL1 由树和背景组成;VOL2 表现一个人。在这个例子中,VOL1 由两个分离的 VOP,即 VOP1 和 VOP3 表示。因此,VOL1 可提供如下意义的基于内容的可分级性,即解码器由于资源限制可选择不解码 VOL1 中的一个 VOP。

VOL2 只包含一个 VOP，即 VOP2。VOP2 可用时间、空间或质量可分级比特流表示。在这种情况下，解码器可再一次决定只解码 VOL2 的低层。图 13.30 中的例子示出了 MPEG-4 支持的基于内容的访问和可分级性的复杂结构。然而，所给的例子也能以简洁的形式用三个 VO 描述。背景、树和人可编码为分离的 VO，每层一个，并且每层由一个 VOP 表示。这些 VOP 被分别编码，并在解码器中合成为一个场景。

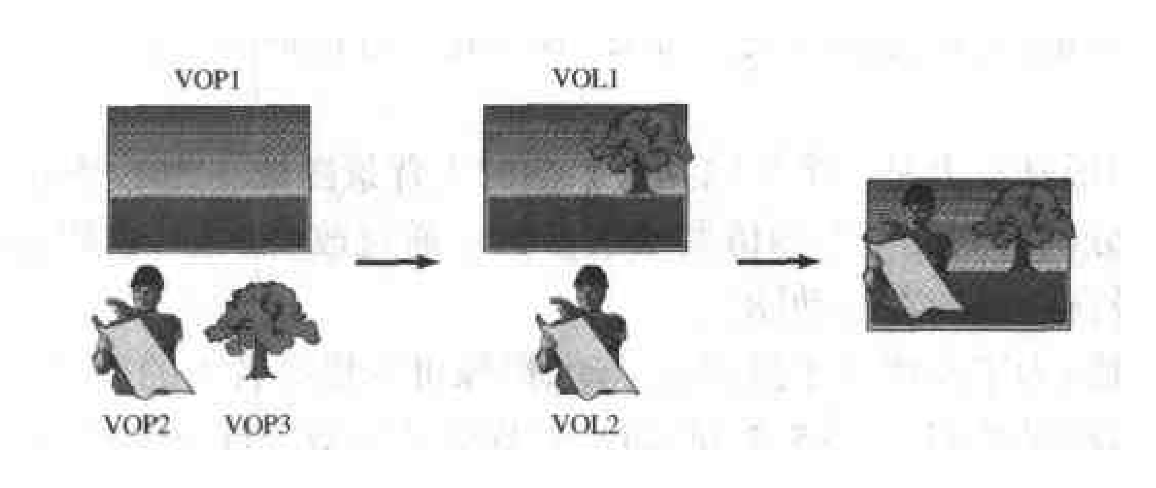


图 13.30 基于对象的编码需要解码器把不同的视频对象平面合成一个场景。视频对象层能够实现基于内容的可分级性

为了弄明白 MPEG-4 视频编码的工作，考虑一 VOP 序列。把 MPEG-1/2 的帧内图像、预测图像以及双向预测图像的概念扩展到 VOP，我们得到 I-VOP，P-VOP 和 B-VOP。如果在一对参考 VOP（I-VOP 或 P-VOP）之间使用两个相继的 B-VOP，则得到的编码结构如图 13.31 所示。

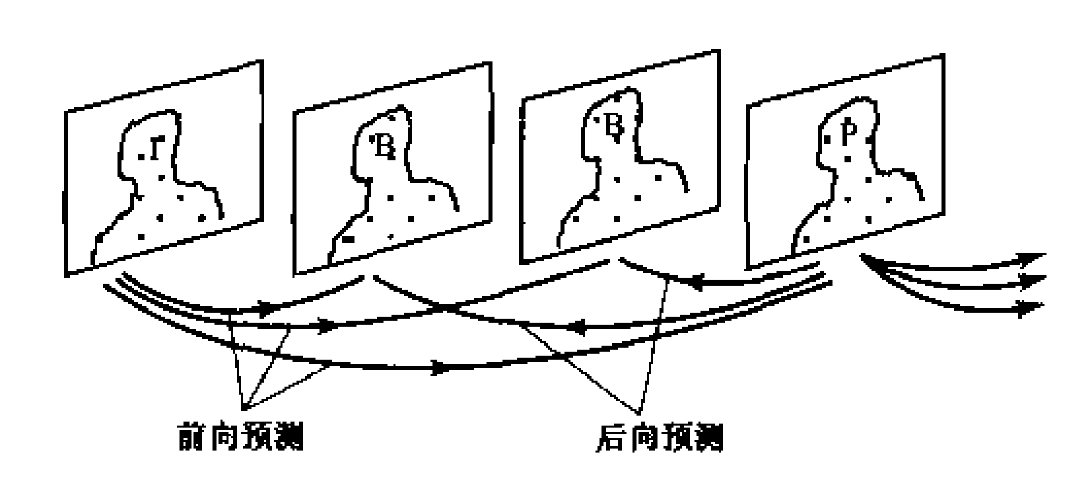


图 13.31 使用 I-VOP，P-VOP 和 B-VOP 的预测结构的例子

高效的编码工具 除了因 MPEG-4 的基于对象的性质所引起的明显改变外，引入了以下的工具以便相对于 MPEG-1 和 MPEG-2 提高其编码效率。

- DC 预测：与 MPEG-1/2 相比改进了 DC 预测。可以选择当前块的前一块或者选择当前块的上一块作为预测器来预测当前 DC 值。
- AC 预测：DCT 系数的 AC 预测在 MPEG-4 中是新的。选择用来预测 DC 系数的块也用于预测一行 AC 系数。如果预测器是前一块，则它的第一列的 AC 系数用于预测和当前块相同位置的 AC 系数。如果预测器是来自前一行的块，则用它来预测 AC 系数的第一行。AC 预测对于具有粗糙纹理、对角边缘或水平以及垂直边缘的块效果不佳。在块级切换 AC 预测的通断是所希望的，但这太昂贵。因此判决是在宏块级做出的。
- 交替水平扫描：这种扫描被添加到 MPEG-2 的两种扫描中（图 13.19）。MPEG-2 的交替扫描在 MPEG-4 中被称为交替垂直扫描。交替水平扫描是由镜像垂直扫描得到的。在

决定 AC 预测的同时选择扫描。在由前一块进行 AC 预测的情况下，选择交替垂直扫描。在由上一块进行 AC 预测的情况下，使用交替水平扫描。AC 预测不与“Z”形扫描相结合。

- 三维 VLC：DCT 系数的编码与 H.263 类似。
- 四个运动矢量：允许宏块的四个运动矢量。做法与 H.263 类似。
- 无约束运动矢量：允许这种模式。与 H.263 相比，可以使用宽得多的 ±2 048 像素的运动矢量范围。
- 子图形：子图形基本上是一个传输到解码器的大背景图像。为了显示，编码器传送把该图像的一部分映射到屏幕上的仿射映射参数。通过改变映射，解码器可以放大和缩小子图形，以及向左或向右摇动[8]。
- 全局运动补偿：为了补偿由于摄像机运动、摄像机变焦或者大的运动物体而引起的全局运动，全局运动是按照公式(5.5.14)的八参数运动模型进行补偿的(见 5.5.3 节)：

$$x' = \frac{ax + by + c}{gx + hy + 1}$$

$$y' = \frac{dx + ey + f}{gx + hy + 1} \tag{13.6.1}$$

  全局运动补偿是改善大的全局运动场景的图像质量的一个重要工具。这些场景是难以用基于块的运动编码的。与具有任意运动的场景不同，人眼能够跟踪全局运动情况下的细节。全局运动补偿有助于改善最挑剔的场景中的图像质量。
- 四分之一像素运动补偿：四分之一像素运动补偿的主要目的是仅以小的语法和计算上的开销来提高运动补偿方案的分辨率，得到更精确的运动描述和较小的需要编码的预测误差。四分之一像素运动补偿将只用于亮度像素；色度像素是以半像素精度进行补偿的。

如已指出的，一些工具类似于在 H.263 中开发的工具。与在 H.263 中一样，MPEG-4 标准描述了重叠块运动补偿。然而这个工具没有包括在任何 MPEG-4 的类中，因为它对于大的图像尺寸计算复杂，并且对于高质量视频提高有限；换句话说，依从 MPEG-4 的解码器不需要实现重叠块运动补偿。

差错复原工具 除了为提高编码效率所开发的工具外，在 MPEG-4 中也定义了一系列工具来增强压缩比特流对传输误差的复原能力。这些将在第 14 章介绍。

13.6.4 基于对象的视频编码

为了使编码视频具有基于物体的功能，MPEG-4 允许传送视频物体的形状。尽管 MPEG-4 没有对定义或分割视频物体的方法标准化，但定义了解码算法，且隐含地定义了描述形状的编码算法。形状是用与亮度信号有相同分辨率的阿尔法图描述的。阿尔法图与亮度图像处于同一位置。MPEG-4 把阿尔法图定义为具有两部分。二进制阿尔法图定义属于物体的像素。在灰度阿尔法图的情况下，我们有另外一个阿尔法图，它用 8 bpp 定义透明度。阿尔法图扩展了宏块。一个宏块的 16×16 的二进制阿尔法图称为一个二进制阿尔法块(BAB)。下面，我们说明 MPEG-4 为基于物体的视频编码所使用的各个工具。

- 二进制形状:基于上下文的算术编码器(如 10.1.1 节所描述的)被用来编码物体的边界块。边界块包含物体像素和背景像素。它与宏块处于同一位置。对于非边界块,编码器只通知宏块是否是物体的一部分。一个阿尔法图序列可以不带纹理进行编码和传输。或者,MPEG-4 使用像填充和 DCT 或 SA-DCT 之类的工具对随物体传输的纹理进行编码。BAB 是用帧内模式和帧间模式编码的。在帧间模式中可以采用运动补偿。形状运动矢量编码用与纹理编码有关的运动矢量作为预测器。
- 填充:为了用基于块的 DCT 编码 BAB 的纹理,背景纹理可设置成任意彩色。在帧内模式中,这种背景彩色对解码图像没有影响,可由编码器进行选择。然而,为了运动补偿,当前块的运动矢量可参考前一个参考图像中的边界块。参考图像的部分背景像素可能位于当前物体的范围内——因此,这些背景图像的值影响预测环。MPEG-4 用填充(如 10.2.1 节描述的)来定义在预测中用到的背景像素。
- 形状自适应 DCT:编码器可选择用 SA-DCT 对 BAB 的纹理进行编码(10.2.2 节)。然而,运动补偿预测图像的填充仍然是需要的。
- 灰度形状编码:MPEG-4 允许传送任意的阿尔法图。因为阿尔法图是用 8 比特定义的,所以用与亮度信号一样的方法对它们进行编码。

图 13.32(a)示出了一个基于物体的 MPEG-4 视频编码器的方框图。MPEG-4 采用两种类型的运动矢量:在图 13.32 中,我们把用来补偿纹理运动的传统的运动矢量称为纹理运动;把描述物体形状位移的运动矢量称为形状运动。形状运动矢量可能与 BAB 相关联。图像分析估计当前 VOPψ_k 相对于参考 VOPψ'_{k-1}的纹理和形状运动。参数编码对参数进行预测编码。参数被传送并解码,而新的参考 VOP 存储在 VOP 存储器中。由编码任意形状的视频物体所引起的复杂度的增加在图 13.32(b)中变得很明显。首先,形状运动矢量和形状像素被编码。形状运动编码器通过分析潜在有损编码的形状参数知道哪些运动矢量要编码。为了纹理预测,参考 VOP 像前面所描述的那样被填充。预测误差用原始形状参数填充,以确定将要填充的区域。然后每个宏块用 DCT 进行编码。

13.6.5　静止纹理编码

MPEG-4 所支持的功能之一是将静止纹理映射到二维或三维表面上。MPEG-4 视频提供了一种编码静止纹理信息的分离模式来支持这种功能。可以想像,与纹理映射的综合场景进行交互操作的应用系统需要连续的可分级性。

为了编码静止纹理图,DWT 编码被选择,因为它在保持良好的编码性能的同时提供了在空间和质量可分级性方面的灵活性(11.3.1 节)。在 DWT 编码中,纹理图图像首先经二维可分离的分解(用 Daubechies 抽头的双正交滤波器)进行分解。接着,最低频带的系数被量化,并用隐含的预测(类似于在帧内 DCT 编码中所用到的)和算术编码进行预测编码。然后较高频带系数用多级量化、零树扫描和算术编码进行编码。产生的比特流是灵活排列的,容易导出多层空间和质量可分级性。

这个算法被推广到编码任意形状的纹理图。为了使形状的扫描行适应 DWT 编码,MPEG-4采用头和尾边界扩展来镜像图像信号(11.3.1 节)。

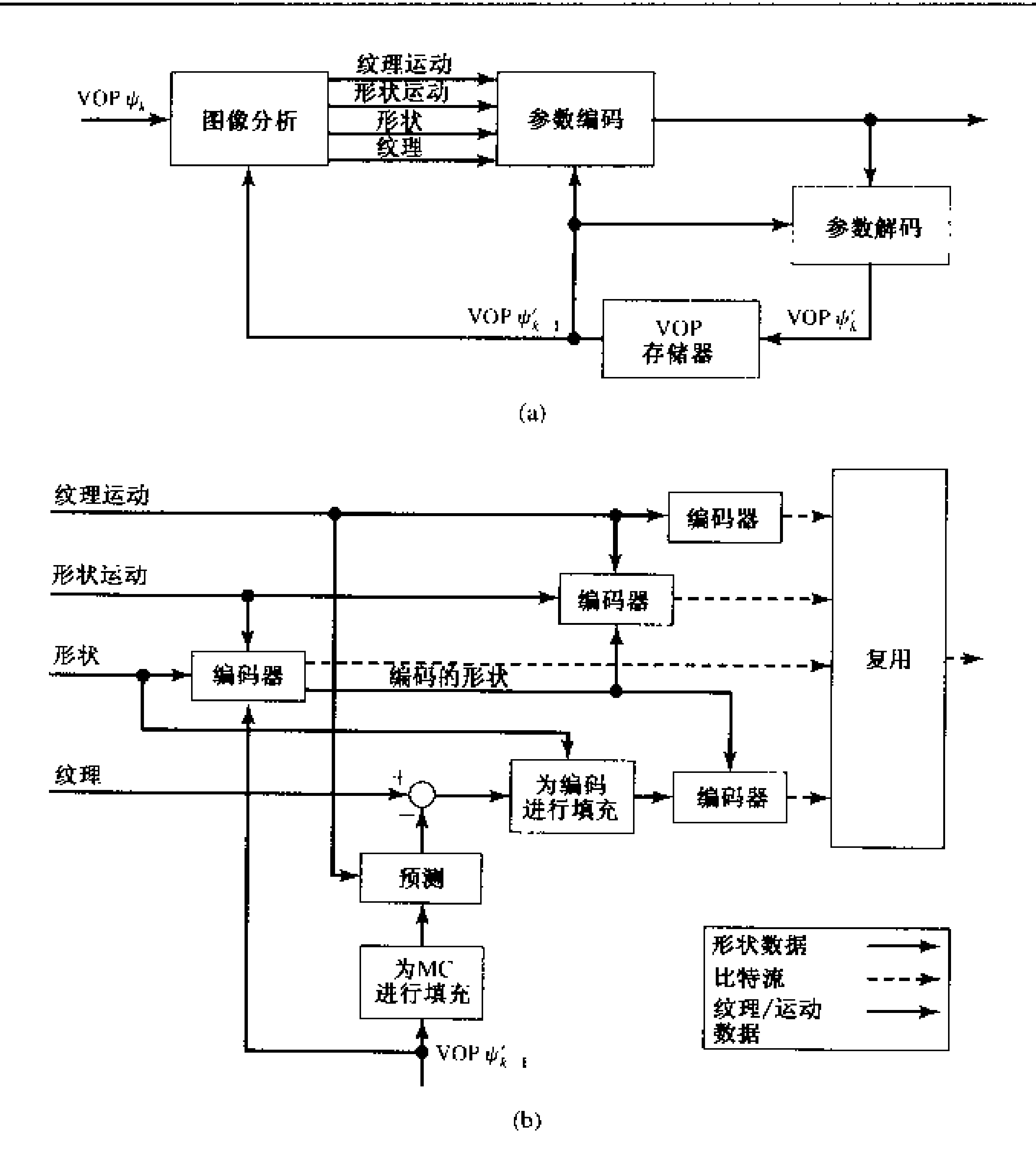


图 13.32 一个编码任意形状视频物体的基于物体的 MPEG-4 视频编码器:(a)总体方块图;(b)参数编码方块图

13.6.6 网格动画

基于网格的物体表示对许多功能是非常有用的,例如动画、内容操作、内容覆盖、自然与合成视频的融合等[68]。

图 13.33 示出了一个网格编码器以及它与纹理编码器的结合。当一个自然的或合成的视频物体初次出现在场景中时,网格编码器为它生成一个二维的基于网格的表示。物体由三角形小块拼接而成,产生一个初始的二维网格(图 13.34)。当 VOP 在场景中移动时,这个初始网格的节点就在二维活动起来。或者,节点的运动也可由另一个信源激活。视频物体的二维运动因此可由网格节点的运动矢量紧凑地表示。通过对应于各个小块的纹理图经仿射变换从一个 VOP 变成另一个 VOP 进行扭曲来实现运动补偿。用于映射到物体网格模型或脸部线框模型上的纹理是由视频或静止图像得到的。尽管网格分析不是标准的一部分,但 MPEG-4 定义了二维网格及其节点运动的编码。另外,把纹理映射到网格上可用 MPEG-4 描述。

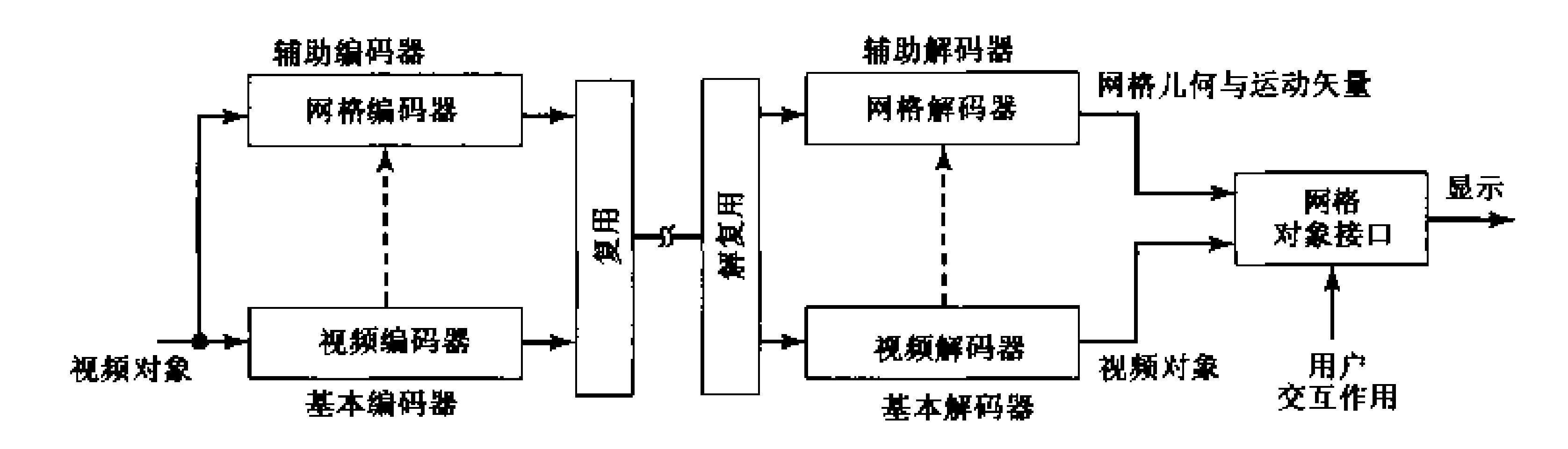


图 13.33　一个支持二维网格物体的编码器/解码器的简化结构。视频编码器提供网格物体的纹理图

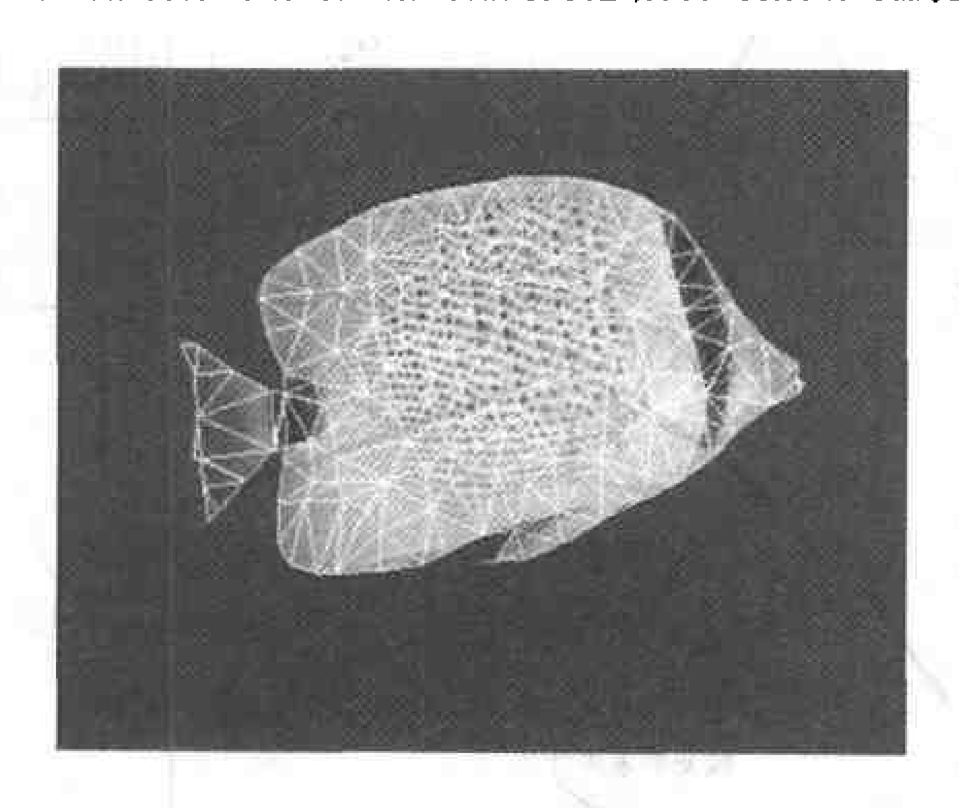


图 13.34　为视频物体"乌鲂"设计的基于内容的网格

13.6.7　脸和身体动画

希望支持脸和身体动画的 MPEG-4 终端包括一个默认的脸和身体模型。MPEG-4 的系统部分提供了脸和身体定义参数(FDP,BDP)来定制脸或身体模型,或用从编码器下载的模型来代替它。场景(包括三维几何体)和脸/身体模型的定义可用 BIFS[23]传送给接收机。图 13.35 示出了一个解码器按照 BIFS 流建立的场景图。身体节点定义身体的位置。它的子 BDP 用具有关节的骨架、表面以及表面特性来描述身体的外形。身体变形表(bodyDefTable)节点描述模型是如何作为身体动画参数的函数变形的。脸节点是身体节点的子节点。它包含脸几何形状并把脸变形定义为脸动画参数(FAP)的函数。MPEG-4 的视频部分定义了如何用 FAP 和身体动画参数(BAP)使这些模型动起来[24]。

图 13.36 示出了左眼眨眼的两个阶段(加上中性阶段),它是用简单的动画结构生成的[68]。图中虚线的半圆示出了当眼睑按照 FAP19 的幅度向下运动时,它的一个顶点的理想运动。在这个例子中,FAP19 的脸变形表用两个线性线段近似目标轨迹,该顶点实际上是随着 FAP19 的增加在这条轨迹上运动的。

身体
BDP
显示的身体
BAP
身体场景图
身体变形表
FBA 解码器
FBA流
关节
关节
BIFS 解码器
BIFS流
IFS
关节
脸
FDP
显示的脸
FAP
脸场景图
脸变形表
IFS
脸变形网格
脸变形变换

图 13.35 一个以 BIFS 流传输的描述人体的场景图。节点身体和脸是用 FBA 流的FAP和BAP激活的。BDP和FDP节点及其子节点描述虚拟的人

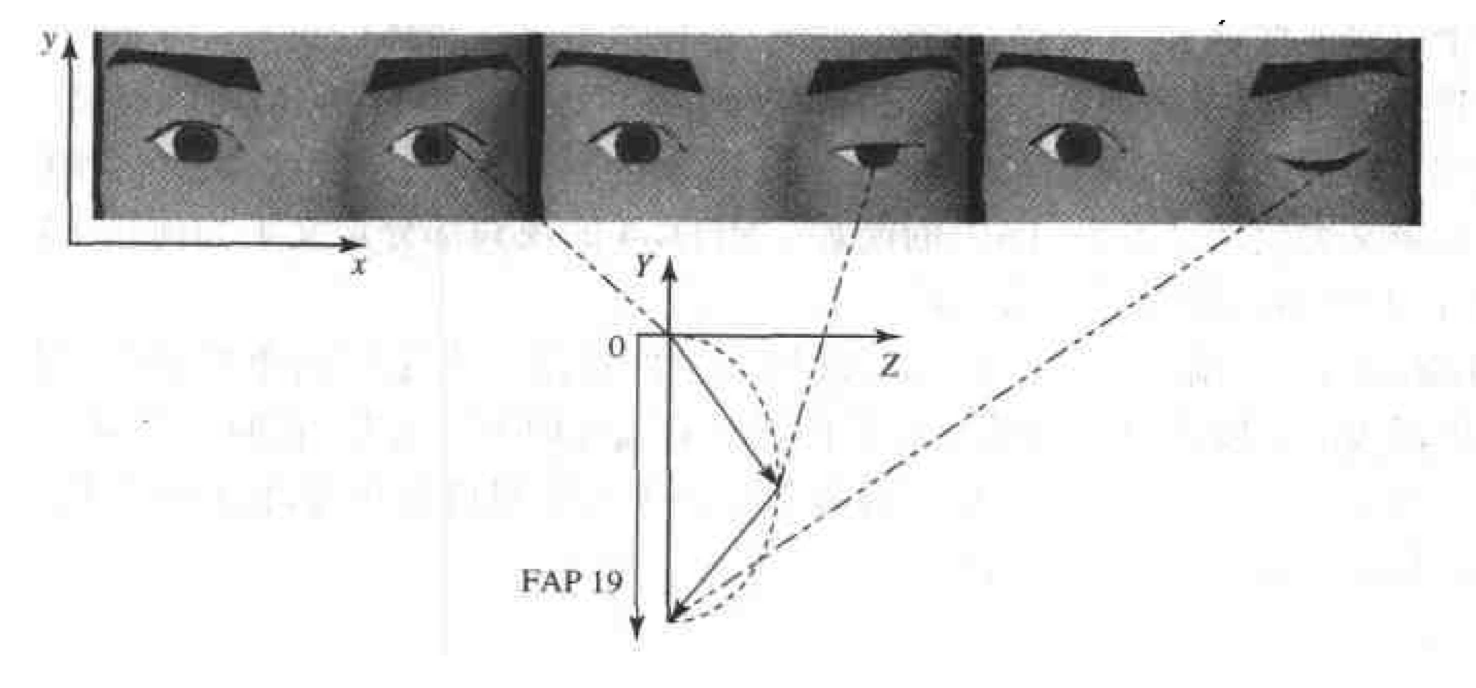


图 13.36 左眼的中性状态(左)和眨眼的两个变形动画阶段(FAP 19)。FAP 定义负 y 方向上眼睑的运动；脸变形表定义 x 和 z 方向上眼睑的一个顶点的运动

脸动画 定义了三组脸动画参数[68]。首先，为低水平的脸动画定义了一组 66 个 FAP，包括头和眼的旋转以及关于嘴、耳、鼻和眼眉变形的特征点的运动（图 10.20）。由于这些参数都是独立于模型的，它们的幅值是按照实际动画模型的比例确定的。第二，为高水平动画定义了一组主要的脸部表情，例如高兴、悲伤、惊讶以及厌恶。第三，对于讲话动画，14 个口型定义了对应于音素的嘴的形状。这些口型被传送到解码器，或者由终端的 TTS 合成器的音素导出。

FAP 被线性量化并用算术编码进行熵编码。或者，一个 16 个 FAP 的时间序列也可以用 DCT 编码。由于高效编码，只需要大约 2 kbps 就可实现生动的脸部表情。

身体动画 BAP 控制身体框架模型中的独立的自由度来生成身体部分的动画[4]。与脸的情况一样，用 BAP 遥控终端中的身体模型可实时地实现栩栩如生的身体场景，而不用每一帧都传送身体的绘画和视频的细节。BAP 将根据不同身体模型的身体姿势和动作，生成相当类似的高水平的结果，也不必向解码器传送模型。在 BAP 集内总共有 186 个预定义的 BAP，另外还有一个包括 110 个用户自定义 BAP 的扩展 BAP 集。每一个预定义的 BAP 对应于连接两个身体部分的关节中的一个自由度。这些关节包括脚趾、踝、膝、臀、脊骨、肩、锁骨、肘、腕和手指。扩展 BAP 是为与身体变形表有关的附加动画特性提供的[1]，例如衣服动画，或人体非骨架部分的身体各部分的动画。

BAP 按照对身体姿势的影响分成若干组。这种分组方案提供了许多优点。首先，它允许我们通过选择一个 BAP 子集来调整动画的复杂度。例如，脊骨中 BAP 的总数是 72，但通过只选择一个预定义的子集可得到相当简单的模型。其次，假设不是所有的运动都包含所有的 BAP，那么可以只传送起作用的 BAP，以便显著减少所需要的比特率。类似于 FAP，BAP 的编码采用算术编码。

语音合成一体化 MPEG-4 认识到 TTS 对于多媒体应用的重要性，为专有的 TTS 提供了一个文本-语音合成器接口。TTS 流包含 ASCII 文本和可选的二进制形式的韵律。解码器按照为 TTS 合成器定义的接口解码文本和韵律信息。合成器生成一个语音样本并把它传送给复合器。复合器为用户提供音频和(如果有必要的话)视频。

图 13.37 示出了语音驱动的脸部动作的结构，可同步地显示合成的语音和讲话的头部。TTS 的第二级输出接口把合成语音的音素以及每个音素的起始时间和持续时间信息送给音素/书签-FAP 转换器。该转换器把音素和定时信息转换成 FAP，脸显示器用它实现脸模型的动画。除了音素外，合成器识别文本中的书签；将相关的非语音 FAP(如高兴)传送给脸显示器。书签的定时信息可由它们在合成语音中的位置导出。由于脸部动画是完全由输入到 TTS 的文本驱动的，所以向解码器传输 FAP 流是可选的。而且实现了同步，因为讲话的头部是由异步的专有 TTS 合成器驱动的。

13.6.8 类

MPEG-4 开发了一个详细的类结构(图 13.38)。尽管可选择一个应用系统实现一个以上的 MPEG-4 类，然而 MPEG-4 终端必须能实现几个类。需要一个对象描述符类，以便能够传输 MPEG-4 流并在终端识别这些流。场景描述类提供把音频、视频或者图形对象组合成一个场景的工具。二维场景描述类只能放置二维视频对象。更高的类可提供更多的功能。必须实现媒体类，以便在终端上显示实际内容。MPEG-4 支持作为媒体的音频、视频和图形。已经定义了几个视频类；我们在这里只列出它们的一个子集，并说明它们的主要功能：

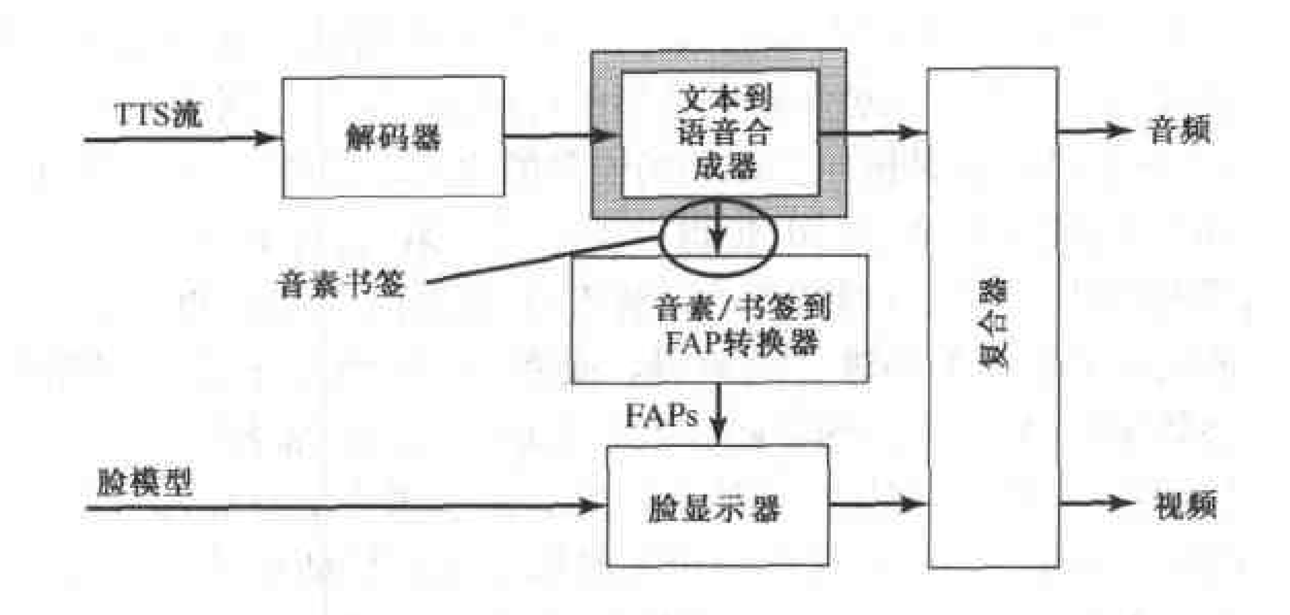


图 13.37 脸部动画的 MPEG-4 结构,允许脸部表情与一个专有文本到语音合成器生成的语音同步

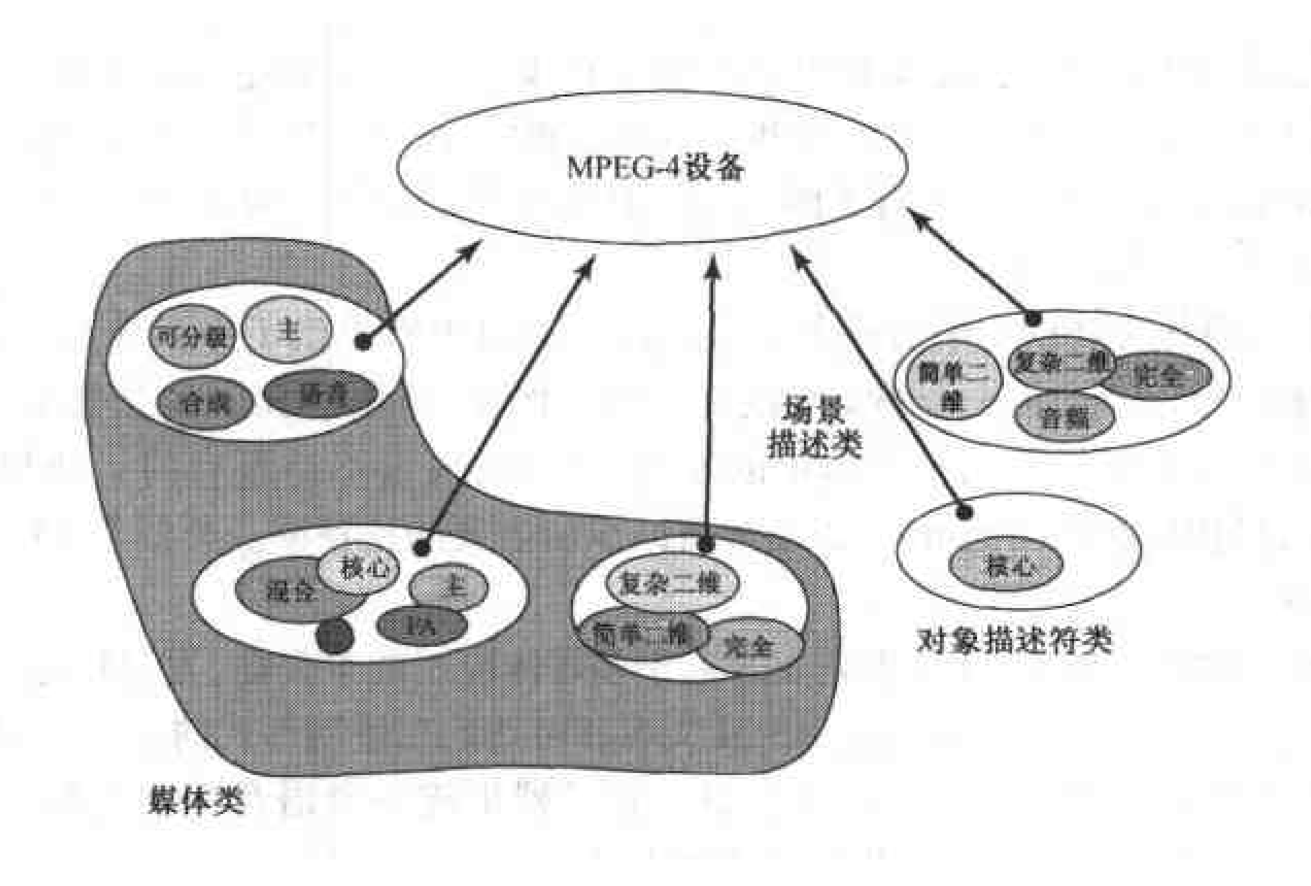


图 13.38 一个 MPEG-4 终端必须为每一个对象描述符类、场景描述类和媒体类至少实现一类。这里没有列出一组内所有的类

- 简单类:简单类是为低复杂度应用需求建立的。第一个用途是移动 (音频)视频业务,第二个用途是把极低复杂度的视频放到因特网上。它最多在一个场景中支持四个矩形对象,这些对象在最低级具有 QCIF 图像的最大尺寸。简单类共分三级,比特率为 64 ~ 384 kbps。它提供如下工具:I-VOP 和 P-VOP、AC/DC 预测、四个运动矢量、无约束运动矢量,片再同步、数据分割以及可逆的 VLC。这个类能解码没有使用任何 H.263 可选附件的 H.263 视频流。
- 简单可分级类:这个类把对 B 帧、时间可分级性和空间可分级性的支持增加到简单类中。它适用于那些由于比特率或解码器资源限制而提供一级以上服务质量的应用系统,如因特网应用和软件解码。
- 先进的实时简单(ARTS) 类:这个类扩展了简单类的能力,用一个后向信道提供矩形视频对象的较复杂的差错保护;这个后向信道把传输差错由解码器通知给编码器,使编码器能为最近编码的图像中受到影响的部分传输帧内模式的视频信息。它适用于实时编码应用系统,如视频电话、电视会议以及远距离观测。

- 先进的简单类：这一类通过增加对四分之一像素运动补偿、全局运动补偿和 B－VOP 的支持，提高了相对于简单类的编码效率。
- 细间隔度可分级类：这一类允许多达八个可分级层，使传送质量能很容易地适应解码资源。当网络支持不同 QoS 级时，较高层可以用比较低层低的优先权传送。增强层是用 DCT 系数的比特面编码器编码的。它可以把简单类或先进的简单类作为基本层使用。
- 核心类：除简单类的工具外，这一类允许可分级的静止纹理、B 帧、二进制形状编码，以及矩形和任意形状物体的时间可分级性。它适用于较高质量的交互式业务，把优良的图像质量与有限复杂度相结合，并且支持任意形状的物体。这个类也能够支持移动广播业务。级 1 的最大比特率是 384 kbps，级 2 是 2 Mbps。
- 核心可分级视频类：这一类把基于物体的 SNR 以及空间和时间可分级性增加到核心类。
- 主类：主类增加了对隔行视频、灰度级阿尔法图以及子图形的支持。主类是为广播业务的需求建立的，可处理逐行和隔行资料。它把最高质量与采用灰度编码的任意形状对象的多功能性结合起来。最高级在 38 Mbps 的最大总码率下接收多达 32 个对象。
- 先进的编码效率（ACE）：这个类以传输比特率低于 1 Mbps 的娱乐视频为目标。然而，根据规定，它加入到主类中，以扩展其比特率范围并增加四分之一像素运动补偿、全局运动补偿和形状自适应 DCT 的工具。这个类不支持子图形。
- 简单演播室类：这个类只对非常高的质量支持 I－VOP，比特率高达 1 200 Mbps。每个 VOP 可以是任意形状的，并有几个与之相关的阿尔法平面。支持 4:0:0，4:2:2 和 4:4:4 格式，在分辨率上适合 HDTV 和数码像机。
- 核心演播室类：这个类把 P－VOP 加到简单演播室类中，从而使这个类更复杂，但是也更有效。

为脸、身体和网格动画定义了更多的类。在写本书时，判断哪些类将最终在产品上实现还为时过早。第一代原型只实现简单类和先进的简单类，而它们的目标应用系统是移动视频通信领域。

13.6.9 视频质量的主观评价

MPEG-4 引入了新的功能（如基于对象的编码），并且声称能提高编码效率。用主观测试的方法验证了其正确性。用户感觉的视频质量是用平均意见分（MOS）度量的[70]。图 13.39 示出了编码效率的主观测试结果，比较了比特率为 384～768 kbps 时 MPEG-4 与 MPEG-1 的视频。这些结果表明 MPEG-4 在这些比特率下性能显著高于 MPEG-1。MPEG-4 视频是用主类的工具编码的（13.6.8 节）。在图 13.40 中，我们看到编码效率的提高是由于 ACE 类的附加工具引起的（13.6.8 节）。768 kbps 码率下的 ACE 类相当于主类在 1 024 kbps 码率下所提供的质量。这使 ACE 类对于通过有线调制解调器或数字用户线路（DSL）传送电影到家中很具有吸引力。进一步的主观测试表明，MPEG-4 中基于对象的功能与基于帧的视频相比不降低编码的视频对象的主观质量——也就是说，因为不编码视频对象以外的像素所节省的比特弥补了形状编码所花费的比特。因此，MPEG-4 的先进工具使基于内容的视频表示不增加视频编码的比特率。

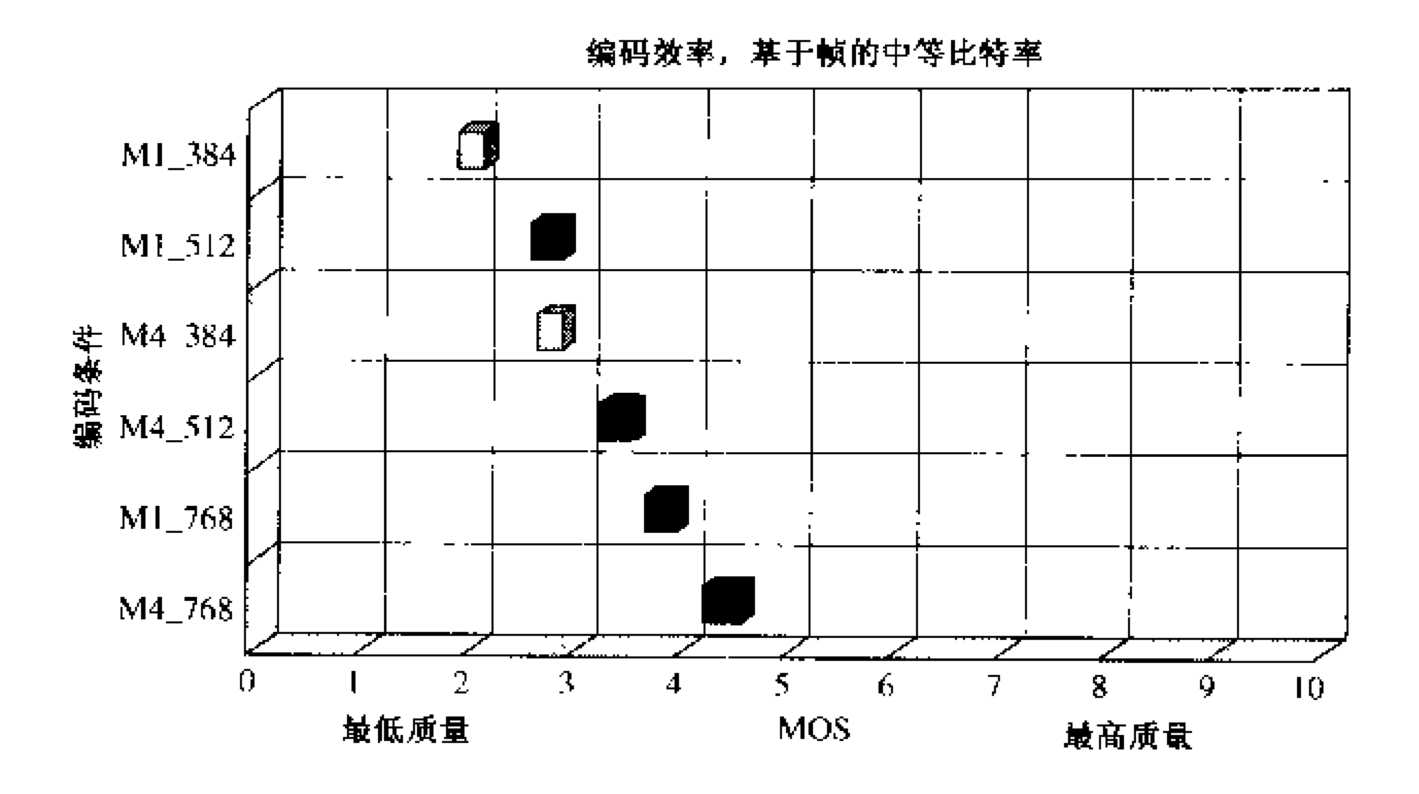


图 13.39 MPEG-4 主类与 MPEG-1 相比的主观质量。M4_x 是工作在 x kbps 速率下的MPEG-4编码器；M1_x是工作在给定速率下的MPEG-1编码器[27]

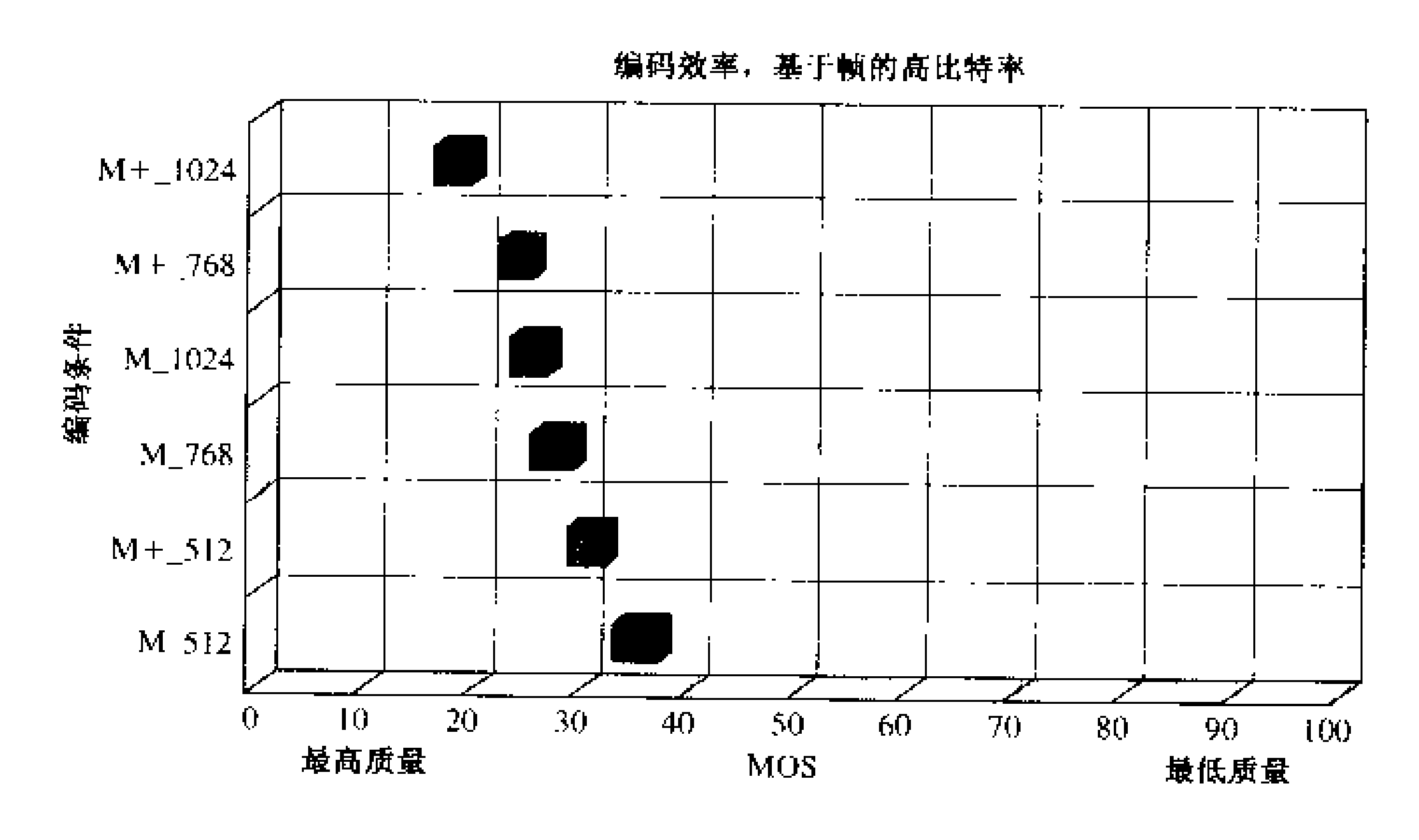


图 13.40 MPEG-4 ACE 与 MPEG-4 主类相比的主观质量。M_x 是按照主类工作在 x kbps 速率下的MPEG-4编码器；M+_x是按照ACE类工作在给定速率下的MPEG-4编码器[26]

13.7 视频比特流语法

如前面提到的，视频编码标准定义视频比特流的语法和语义，而不是实际的编码方案。它们也规定如何解析和解码比特流以生成解压缩的视频信号。为了支持不同的应用，语法必须是灵活的。这是通过一个具有不同层的分层结构实现的，每一层以一个头开始。每层执行不同的逻辑功能（表 13.6）。大多数的头可在比特流中被惟一地识别，因为它们以一个起始码开始；这个码是一长串零（MPEG-2 中是 23 个零），后跟一个 1 和一个起始码识别符。图 13.41 可视化了 MPEG-2 的分层结构。在以下清单中，我们描述了分层比特流结构的元素：

1. 序列：视频序列以序列头开始，且可以包含附加序列头。它包括一个以上的图像组，以一个序列结束码作为结束。序列头及其扩展包含基本的参数，如图像尺寸、图像幅型比、图像速率以及其他全局参数。视频对象层(VOL)的头具有相同的功能，然而它携带 MPEG-4 解码器所需要的附加信息，以便把任意形状的视频对象组合为一个要显示的视频序列。

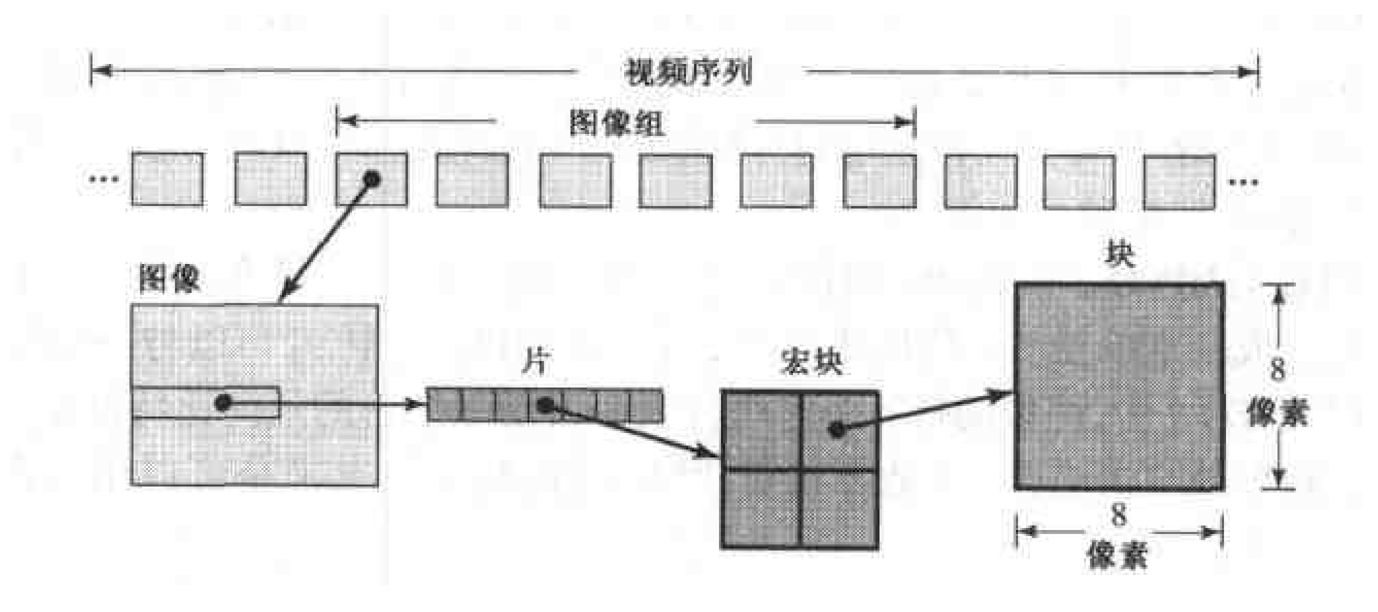


图 13.41　MPEG-2 比特流从视频序列层到块级的形象化的分层结构，示出的是亮度分量。每一层也有两个与它相联系的色度分量

表 13.6　用于各种视频编码标准中的语法层次*

语法层	功能	标准
序列(SC)	定义整个视频序列	H.261/3，MPEG-1/2
VOL(SC)	定义整个视频对象	MPEG-4
GOP(SC)	能够随机访问视频流	MPEG-1/2
GVOP(SC)	能够随机访问视频流	MPEG-4
图像(SC)	基本的编码单元	H.261/3，MPEG1/2
VOP(SC)	基本的编码单元	MPEG-4
GOB(SC)	再同步、刷新和图像中的差错恢复	H.261/3
片(SC)	再同步、刷新和图像中的差错恢复	MPEG-1/2
视频包(SC)	再同步和图像中的差错恢复	MPEG-4
宏块	运动补偿和形状编码单元	H.261/3，MPEG-1/2/4
块	变换和补偿单元	H.261/3，MPEG1/2/4

* 每一层以头作为开始。语法层中的 SC 表示该层的头以起始码作为开始。

2. 图像组：GOP 是一个头后跟一系列图像，意图是允许随机访问序列、快速搜索和编辑。因此，GOP 中的第一个图像是帧内编码图像(I 图像)。它的后面是前向预测编码图像(P 图像)和可选的双向预测图像(B 图像)的一个排列。GOP 头也包含用于同步和编辑的时间码。GOP 是编辑和随机访问的基本单元，因为它的编码不依赖于前面的 GOP 和后继的 GOP。在 MPEG-4 中，GOP 的功能是由视频对象平面组(GOVOP)提供的。由于 H.261 和 H.263 主要是为交互式应用设计的，所以它们不用 GOP 的概念。然而，编码器可随时选择传送一个 I 图像，从而实现随机访问和简单的编辑。
3. 图像：图像是视频序列的基本编码单元。图像由表示亮度(Y)和两个色度(Cb 和 Cr)值的三个矩形矩阵组成。图像头指明图像类型(I，P，B)、图像结构(场/帧)以及可能的其他参数(例如运动矢量范围)。VOP 是 MPEG-4 的基本编码单元。它包含视频对象边框

的尺寸。

每个标准都把图像分成宏块组。H.261 和 H.263 用固定的宏块结构，而 MPEG-1/2 允许灵活的结构。MPEG-4 把可变数量的宏块安排成一组。

4. GOB：H.261 和 H.263 把图像分成 GOB，每个 GOB 包括三行宏块，每个 GOB 行中有 11 个宏块。GOB 头定义 GOB 在图像内的位置。可以为每个 GOB 定义一个新的量化器步长。在处理差错中 GOB 是很重要的。如果比特流含有一个差错，解码器可跳到下一个 GOB 的开始，从而把比特差错的扩散限制在当前帧的一个 GOB 之内。然而，当预测下一帧时，差错扩散可能发生。
5. 片：MPEG-1，MPEG-2 和 H.263 附件 K 把 GOB 的概念推广到可变结构。片把几个相继的宏块组成一个单元。片的尺寸可以变化。在 MPEG-1 中，片可以与一幅图像一样大。然而在 MPEG-2 中，图像的每一个宏块行至少启动一个新的片。比特流中具有较多的片允许较好的差错隐藏，但是要花费比特，而这些比特本来是可以用来提高图像质量的。
6. 视频包头：MPEG-4 所用的视频包方法是基于在整个比特流中提供周期的再同步标志的。换句话说，视频包的长度并不基于宏块的数量，而是基于包中所含的比特数。如果当前视频包中所含的比特数超过编码器所规定的阈值，那么在下一个宏块的开始处就要建立一个新的视频包。这样，与较刚性的片和 GOB 结构的情况相反，传输差错对具有较高活动性的区域的破坏比对静止区域要小。视频包头携带位置信息并重复图像头的信息，这些信息对于解码视频包是必要的。
7. 宏块：宏块是图像中的一个 16×16 像素的块。用 4:2:0 格式时，每个色度分量的垂直和水平分辨率为亮度分量的一半。因此，一个宏块包括 4 个 Y 块、一个 Cr 块和一个 Cb 块。它的头携带相对位置信息、量化器尺度信息，MTYPE 信息（I，P，B）以及一个 CBP，它指明宏块的 6 个块中哪一个是编码的（和如何编码的）。与其他的头一样，其他的参数可能出现也可能不出现在头中，这取决于 MTYPE。由于 MPEG-4 也需要编码视频对象的形状，它用二进制阿尔法块扩展了宏块，阿尔法块为宏块中的每个像素定义了是否属于 VO。在灰度阿尔法图的情况下，宏块也包含编码的阿尔法图的四个块。
8. 块：块在标准化的视频编码算法中是最小的编码单元。它由 8×8 像素组成，且可以是三种类型中的一种：Y，Cr 或 Cb。块的像素可由它们的 DCT 系数表示，通过哈夫曼码记录在下一个非零系数前零的个数和这个系数的幅值。

比特流中的各种头允许解码器从比特流的差错中恢复，并且一旦接收到起始码就立即开始解码。比特流语法的设计确保不存在合法的码字组合可以仿效起始码。经常在语法中用一个反比特来避免这种起始码的仿效。在标准中没有定义当接收到错误的比特流时解码器的行为。不同的解码器可以具有不同的行为：一些解码器会崩溃，并要求重新启动终端；一些解码器则在一幅图像内恢复，还有一些解码器则要等到下一个 I 帧出现后才能再次开始解码。

13.8 采用 MPEG-7 的多媒体内容描述

在普遍存在的视频应用中，语言、音频、视频、图像和视频序列的索引和搜索变得非常重

要。MPEG-7 是正在进行的一种致力于音视频文件内容描述的标准[32,64]。从原理上说，MPEG-1/2/4 是为表示信息本身设计的，而 MPEG-7 则打算表示关于信息的信息。从另一个角度看：MPEG-1/2/4 使内容成为可利用的，而 MPEG-7 使你能够找到所需要的内容[64]。MPEG-7 的意图是为其他的 MPEG 标准提供补充功能：表示关于内容的信息，而不是内容本身("关于比特的比特")。MPEG-4 能把有限的文本的元信息附加到它的流中，而 MPEG-7 标准提供全套的索引和搜索能力，使得我们不仅能用文本键而且也能用像彩色直方图、运动轨迹等键来搜索视频序列。MPEG-7 于 2001 年成为国际标准。

在本节中，首先给出被 MPEG-7 标准化的元素的概述；然后说明多媒体描述方案，集中于内容描述。我们解释 MPEG-7 如何分解一个音视频(AV)文件以达到结构和语义描述。最后，我们说明用于这些描述中的视频描述符。描述符和以下给出的描述方案假设语义上有意义的区域和对象可以被分割，而且形状、运动参数甚至这些区域和对象的语义标签可以被精确抽取出来。然而，我们注意到产生这样的信息仍是一个有待解决的问题，而且可能需要手工帮助。MPEG-7 标准只定义可用来确定这些信息的语法，而不定义用来抽取它们的算法。

13.8.1 概述

MPEG-7 标准[32]的主要元素是：

- 描述符(D)：MPEG-7 描述符是为了表示特性而设计的，包括低级的音视频特性，高级的语义对象、事件和抽象概念的特性，关于存储媒体的信息，等等。描述符定义每个特性表示的语法和语义。
- 描述方案(DS)：MPEG-7 DS 在 MPEG-7 描述符的基础上进行扩展，把单个描述符以及其他更复杂结构内的 DS 组合起来，并且定义了组合的描述符和 DS 之间的关系。
- 描述定义语言(DDL)：DDL 是能够生成新的 DS 和可能的新描述符的语言。它也允许扩展和修改现存的 DS。已选择 XML 图解语言为 DDL 提供基础。
- 系统工具：这些是为有效传输和存储 MPEG-7 的描述，为实现内容与描述之间的同步以及为管理和保护知识产权所需要的工具。

13.8.2 多媒体描述方案

在 MEPG-7 中，把 DS 分类为适合于音频或视频领域的或适合于多媒体描述的。多媒体 DS 按照它们的功能分成以下几类(见图 13.42)：

- 基本元素：这些涉及基本数据类型、数学结构、图解工具、链接和媒体局部化工具以及基本 DS，基本 DS 是更复杂 DS 的基本成分。
- 内容描述：这些 DS 描述 AV 文件的结构和概念形态。
- 内容管理：这些工具规定关于存储媒体、AV 文件的创建及应用的信息。
- 内容组织：这些工具通过分类、通过定义和收集 AV 文件以及通过建立模型来处理内容的组织。
- 浏览与访问：这些包括浏览总结以及把同一个 AV 内容进行变化以适应客户终端的性能、网络条件或用户选择。
- 用户交互：这些 DS 规定属于多媒体资料消费的用户选择。

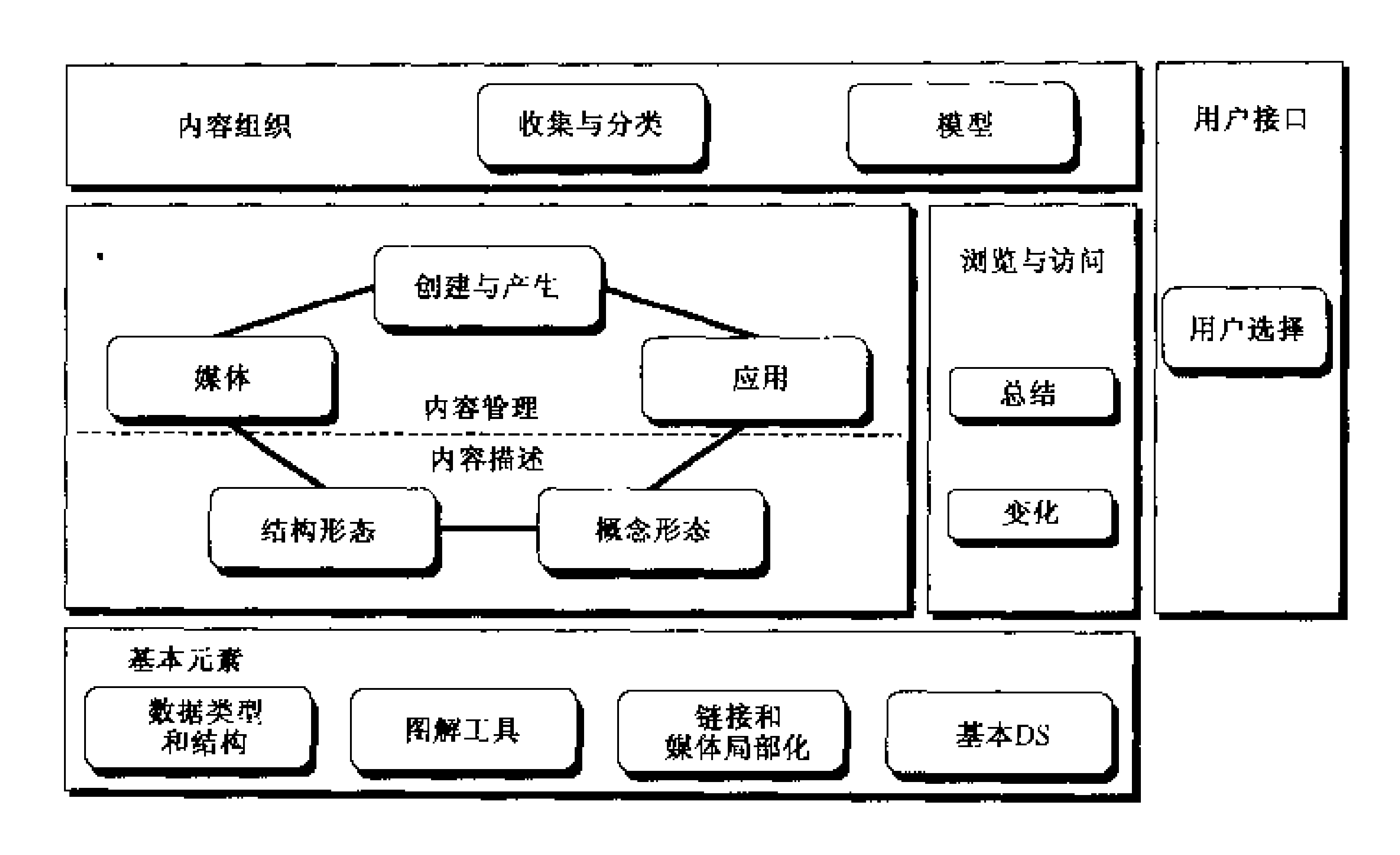


图 13.42 MPEG-7 多媒体描述方案概览

内容描述 下面我们简单说明内容描述的 DS。更详细的信息可在参考文献[29]中找到。为内容描述开发的 DS 分为两类:描述 AV 文件结构形态的 DS 和描述概念形态的 DS。

结构形态:这些 DS 根据段和区域描述 AV 文件的语法结构。AV 文件(例如具有音轨的视频节目)分成段的层次结构,称为段树。例如,整个文件被分段成几个故事单元,然后再把每个故事单元分成不同的场景,最后把每个场景分裂成许多摄影镜头。树的每一级上的段可再进一步分成视频和音频段,分别对应视频帧和音频波形。除了使用包含一组完整视频帧的段外(它们在时间上可能是不邻接的),也可以抽取出静止的或运动的区域。区域可递归地分为子区域,形成区域树。图 13.43 的左边图示了段树的概念。

概念形态:这些 DS 根据事件、对象以及其他的抽象符号描述 AV 文件的语义内容。语义 DS 描述文件中发生的事件和对象,并且为它们贴上相应的"语义标签"。例如,事件类型可能是新闻广播、体育比赛等。对象类型可能是人、汽车等。与结构描述一样,MPEG-7 也用层次分解来描述 AV 文件的语义内容。事件能进一步分解成许多子事件,以形成事件树(图 13.43 的右边)。对象树也可类似地形成。事件-对象关系图描述事件与对象之间的关系。

结构与语义 DS 之间的关系:事件一般与段相关联,而对象与区域相关联。每个事件或对象可以在文件中发生许多次,它们实际的位置(段或区域)是由一系列链接描述的,如图 13.43 所示。从这个意义上,段树和区域树所表示的语法结构像一本书开始处的目录表,而事件树和对象树所表示的语义结构类似于书末尾的索引。

13.8.3 视觉描述符和描述方案

对于段或区域树任意一级上的每个段或区域,一组音频和视频的描述符和 DS 被用来表征这个段或区域。在本节中,我们简单地介绍用于描述视频段或对象的彩色、纹理、形状、运动和位置的视频描述符和 DS。在参考文献[28,33]中可找到更完整的描述。

彩色 这些描述符描述在视频段、运动区域或静止区域中的彩色分布。

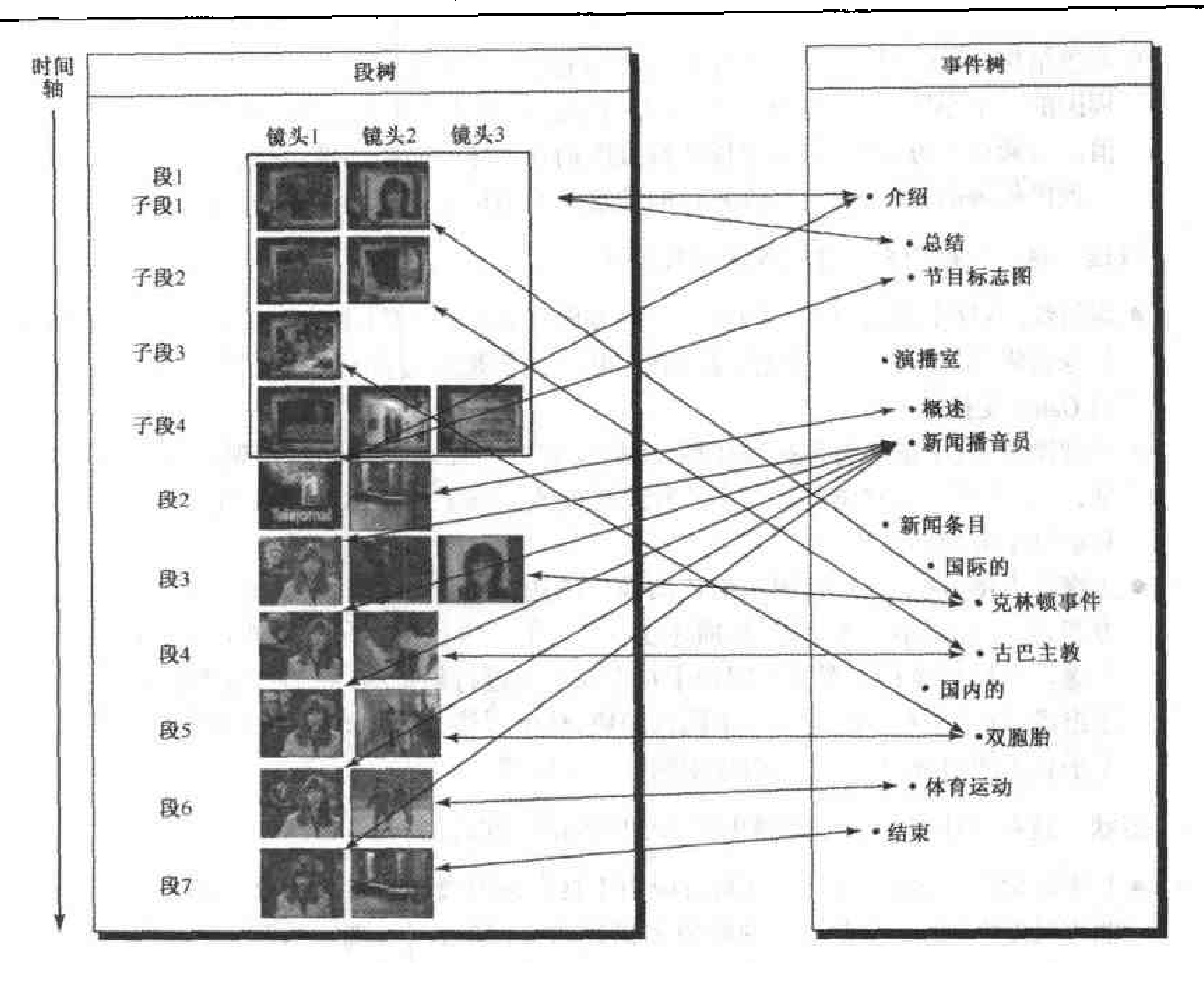


图 13.43 基于段和事件树的 AV 文件的描述(本例是一个新闻节目)。段树就像一本书开始处的目录表,而事件树类似于书末尾的索引

- 彩色空间:定义了五种彩色空间:RGB,YCrCb,HSV,HMMD 和单色。或者,我们可由 RGB 坐标规定一个任意的线性变换矩阵。
- 彩色量化:这个描述符用来确定量化参数,包括量化级数和每个彩色分量的起始值。只考虑均匀量化。
- 主导彩色:这个描述符规定段中的主导彩色,包括主导彩色数,指示主导彩色的空间凝聚性的值(即主导彩色是在段中分散的还是形成一丛),以及对于每种主导彩色取该彩色的像素的百分比,彩色的值和它的方差。
- 彩色直方图:彩色直方图定义在 HSV 空间。不用彩色直方图本身,而是对直方图进行哈尔变换,并且取决于可利用的比特率,用可变的精度指定哈尔系数。可规定几类直方图。普通彩色直方图称为可分级彩色,它包括每个量化彩色在段或区域的所有像素中所占的百分比。GoF/GoP 彩色指的是一组帧或图像上传统直方图的平均值、中值或交集 (每种彩色的最小百分比)。
- 彩色布局:这个描述符用于粗略描述图像的彩色模式。图像被缩小为 8×8 的块,每个块用它的主导彩色表示。然后用 DCT 对缩小图像中的每一种彩色分量(Y/Cb/Cr) 进行变换,并指定前几个系数。

- 彩色结构:这个描述符意在捕捉具有相同彩色的像素的空间凝聚性。只要在每个像素周围的一个小区域内至少有一个具有该彩色的像素(称为结构元素),就增加彩色计数值。与彩色直方图不同,这个描述符可区别开这样的两幅图像:它们对于所给定的彩色呈现出相等的数量,但具有该彩色的像素组的结构在两幅图像中是不同的。

纹理 这一类被用来描述图像的纹理模式。

- 均匀纹理:这个描述符用来确定在不同方向和频带(尺度)中的能量分布。前两个分量是像素强度的均值和标准差。后面的 30 个分量是通过具有 6 个方向区和 5 个尺度带的 Gabor 变换获得的。
- 纹理浏览:这个描述符根据规律性、粗糙度和方向性来确定纹理外观,它符合描述的类型,一个人可以用它浏览或检索一种纹理模式。除了规律性外,可确定两个主要的方向和沿每个方向的粗糙度。
- 边缘直方图:这个描述符用来描述图像中的边缘方位分布。可指定三种类型的边缘直方图,每一种类型有 5 个记录,描述在 4 个可能的方位中方向边缘的百分比以及非方向边缘。全局边缘直方图是在图像中每个像素上进行累积的;局部直方图由 16 个子直方图组成,每个子直方图对应一个图像的块,整个图像分成 4×4 个块;半全局直方图由 13 个子直方图组成,每个对应图像中的一个子区域。

形状 这些描述符用来描述静止和运动区域的空间几何。

- 基于轮廓的描述符:这个描述符可应用于具有封闭边界的二维区域。MPEG-7 已选择用曲率尺度空间(CSS)表示中的峰值来描述边界[56];已发现它反映了人对形状的感觉——在这种表示中相似的形状具有相似的参数。边界的 CSS 表示是通过用平滑滤波器递归地模糊原始边界,沿每条已滤波的曲线计算曲率,最后在连续模糊后确定曲率的过零位置来获得的。描述符规定了在 CSS 中曲率峰的数目、边界的全局离心率和圆度、原型曲线(它是导致 CSS 中最高峰的曲线)的离心率和圆度,原型滤波器以及残留峰的位置
- 基于区域的形状描述符:这个描述符利用了组成形状的所有像素,因此可以描述任意的形状——不仅是具有单连通区域的简单形状,也可以是由几个不相连的区域组成的复杂形状。具体地说,由阿尔法图表示的原始形状被投影到角射线变换(ART)的基函数上;描述符包括 ART 系数的 35 个归一化的和量化的幅值。
- 三维形状:这个描述符提供了三维网格模型的内在描述。它利用了三维表面的一些局部属性。为了导出这个描述符,在网格表面的每点处计算所谓的形状索引,它取决于每点的主曲率。描述符规定了形状谱,形状谱是在整个网格上计算的形状索引的直方图。直方图中的每个记录实际上规定了特定区间中具有形状索引的所有三维网格表面区域的相对面积。另外,描述符包括没有定义形状索引的网格的平坦表面区域的相对面积以及所有单个多边形元素的相对面积,这些元素是不可能可靠估计形状索引的区域。

运动 这些描述符描述视频段或运动区域以及全局摄像机运动的运动特性。

- 摄像机运动:考虑了 7 种可能的摄像机运动:摇、跟(水平移动)、倾、吊(垂直移动)、变焦、推(沿光轴移动)、滚(绕光轴旋转)(见图 5.4)。每一种运动可能有两个运动方向。

对于每一种运动类型和方向，规定了出现（即持续时间）、速度和运动量。最后一项度量由特定的运动引起的覆盖或露出的面积。

- 运动轨迹：在所选择的采样时间，这个描述符根据某些关键点的二维或三维坐标规定非刚性运动物体的轨迹。对于每一个关键点，用一个指定的内插函数（线性的或抛物线的）内插出两相邻采样时间之间的轨迹。
- 参量物体运动：这个描述符用来规定刚性运动物体的二维运动。包括 5 种运动模型：平移、旋转/缩放、仿射、平面透视和抛物线。平面透视和抛物线运动分别参考公式(5.5.14)定义的投影映射和公式(5.5.19)定义的双二次映射。除了模型类型和模型参数外，还必须指定坐标原点和时间间隔。
- 运动活动性：这个描述符用于描述视频段（一般是在镜头级）的活动强度和范围。可指定 5 种属性：(1)活动性的强度，由运动矢量幅度的标准偏差度量；(2)活动性的方向，规定所有运动矢量的主导方向或平均方向；(3)活动性的空间分布，由运动幅度低于平均幅度的块的游程导出；(4)活动性的空间定位；(5)活动性的时间分布，用镜头中单个帧上量化的活动级的直方图来描述。

位置　这些描述符和 DS 用来描述静止或运动区域的位置。

- 区域定位符：这个描述符用简单且可分级的边框或多边形表示来确定区域位置。
- 空间－时间定位符：这个 DS 描述运动区域。它把整个区域的持续时间分解成几个子段，每个段用段开始处的区域（称为参考区域）的形状以及该区域与下一段的参考区域之间的运动来确定。对于非刚性物体，开发了一个图轨迹 DS，它通过限定矩形、椭圆或多边形定义参考区域，并且用运动轨迹描述符确定参考区域之间的运动，该描述符确定在相继采样时间上所选择的关键点的坐标。对于刚性区域，使用参数轨迹 DS，它用区域定位器描述符来确定参考区域，并且用参量物体运动描述符来描述运动。

13.9　小结

视频通信需要标准化以便生产价位合理、协同工作并迎合大市场的设备。个人视频电话是数字视频压缩标准所瞄准的第一个应用。H.261 发布于 1990 年——距 Jules Verne 写下视频电话思想有 101 年，比他预测的提前了 899 年[69]。相继出现的重要视频压缩标准 H.263，MPEG-1，MPEG-2 和 MPEG-4 分别于 1993 年、1995 年、1995 年和 1999 年建立。

H.261 和 H.263 标准只描述视频压缩，而 MPEG-1/2/4 标准也描述音频表示以及能联合传输音视频信号的系统表示。H.261 是具有整数像素运动补偿的基于块的混合编码器。H.261 的主要应用是对以 64 kbps 和 2 Mbps 之间的速率在 ISDN 线上传输的视频会议进行视频编码。H.263 扩展了 H.261，并且增加许多特性（包括半像素运动补偿），因此使编码视频能够以低于 56 kbps 的速率在模拟电话线上传输。

MPEG-1 也源于 H.261。它增加了半像素运动补偿、B 图像的双向预测以及其他的改善，以便满足大约 1.2 Mbps 速率的编码视频以及 CIF 分辨率的 CD-ROM 上的消费视频的需求。MPEG-2 是第一个能以全电视和 HDTV 分辨率编码隔行视频的标准。它扩展了 MPEG-1，为隔行视频引入了新的预测模式。它主要应用于大约 4 Mbps 速率的 TV 广播以及 15 Mbps 的高质

量视频方面。基于 MPEG-2 和 H.263 的 MPEG-4 视频是最新的视频编码标准，它引入基于对象的功能，不仅用运动和纹理，而且用形状来描述视频对象。形状信息与亮度信号同位，并且用基于上下文的算术编码器进行编码。

MPEG-2 和 MPEG-4 定义类，要求解码器实现标准定义的工具的一个子集。这使得能够建立标准的解码器，这些解码器是面向特定应用领域而配置的。

MPEG-1/2/4 标准的开发使音视频数据的交换成为可能，而 MPEG-7 的目的是使搜索和浏览这种数据成为可能。MPEG-7 可独立于其他 MPEG 标准使用——MPEG-7 描述甚至可以加入模拟电影中。MPEG-7 描述可用来提高以前的 MPEG 标准的功能，但不会代替 MPEG-1，MPEG-2或 MPEG-4。

由于终端的计算能力在逐年增加，标准化组织试图提高它们的标准。ITU 目前正在制定视频编码标准 H.26L，承诺对于相同的比特率比 H.263 和 MPEG-4 改善 1dB 以上，或者当编码视频速率高于 128 kbps 时，对于相同的图像质量，比特率减少 20% 以上。

13.10 习题

1. 在标准的内容中，你知道关于兼容性的哪些种类？
2. 哪些是 H.261 视频编码器的最大计算强度的部分？哪些是解码器的最大计算强度的部分？
3. 环路滤波器是什么？为什么只有 H.261 标准实现它呢？
4. H.263 比 H.261 提高编码效率的工具是哪些？
5. 按照改进的 PB 帧模式，MPEG-1 B 帧与 H.263 PB 帧之间的主要区别是什么？
6. H.323 和 H.324 标准的目的是什么？
7. 为什么 MPEG-2 具有一种以上的扫描模式？
8. I 帧的知觉量化对编码图像的 PSNR 有何影响？知觉量化是如何影响图像质量的？选择加权矩阵系数的一个好的指导原则是什么？
9. 解释 MPEG-2 中类和级的概念。
10. 在商用产品中使用 MPEG-2 的哪个类？为什么也存在其他的类？
11. MPEG-2 支持哪种类型的可分级性？
12. 什么是漂移？它什么时候出现？
13. 讨论 H.261，H.263 和 MPEG-1/2/4 提供的差错复原工具。为什么 MPEG-4 差错复原工具最适于有损传输信道？
14. 在功能和编码效率方面，MPEG-1 层 III 音频编码与 MPEG-2 AAC 有什么区别？
15. MPEG-4 允许编码形状信号。在二进制形状的情况下，有多少个块与宏块相联系？它们的尺寸多大？关于灰度形状编码的情况又怎样？
16. 为什么按照 ACE 类的 MPEG-4 视频优于 MPEG-1 视频？
17. MPEG-4 终端的哪个部分没有被标准化？
18. 为什么视频比特流包含起始码？
19. 什么是元信息？
20. 哪个标准用小波编码器？目的是什么？

21. 为什么如 MPEG-4 中所定义的 FAP 对内容生成很重要?
22. 在语音合成器与讲话的脸之间是如何达到同步的?
23. MPEG-4 网格动画的功能和目的是什么?
24. 视频索引和检索的困难是什么? 一个标准化的内容描述接口,例如 MPEG-7,如何能简化视频检索?
25. MPEG-7 中的段树是如何描述视频序列的语法结构的? MPEG-7 中的事件树是如何描述视频序列的语义结构的? 它们的关系是什么?
26. MPEG-7 开发的视频描述符是什么? 假定这些描述符被添加到一个大的视频数据库中的每个视频序列中。说明使用它们来检索某类序列的方法。

13.11 文献目录

[1] Avaro, O., A. Elftheriadis, C. Herpel, G. Rajan, and L. Ward. MPEG-4 systems: Overview. In A. Puri and T. Chen, eds., *Multimedia Systems, Standards, and Networks*. New York: Marcel Dekker, 2000, 331–65.

[2] Basso, A., M. R. Civanlar, and V. Balabanian. Delivery and control of MPEG-4 content over IP networks. In A. Puri and T. Chen, eds., *Multimedia Systems, Standards, and Networks*. New York: Marcel Dekker, 2000, 501–23.

[3] Brandenburg, K., O. Kunz, and A. Sugiyama. MPEG-4 natural audio coding. *Signal Processing: Image Communication* (2000), 15(4-5):423–44.

[4] Capin, T. K., E. Petajan, and J. Ostermann. Efficient modeling of virtual humans in MPEG-4. In *IEEE International Conference on Multimedia and Expo (ICME2000)*, 2:1103–6, New York, 2000.

[5] Chen, T., Emerging standards for multimedia applications. In L. Guan, S. Y. Kung, and J. Larsen, eds., *Multimedia Image and Video Processing*. CRC Press, 2000, 1–18.

[6] Chen, T., G. J. Sullivan, and A. Puri. H.263 (including H.263++) and other ITU-T video coding standards. In A. Puri and T. Chen, eds., *Multimedia Systems, Standards, and Networks*. New York: Marcel Dekker, 2000, 55–85.

[7] Chiariglione, L., Communication standards: Götterdämmerung? In A. Puri and T. Chen, eds., *Multimedia Systems, Standards, and Networks*. New York: Marcel Dekker, 2000, 1–22.

[8] Dufaux, F., and F. Moscheni. Motion estimation techniques for digital TV: A review and a new contribution. *IEEE* (June 1995), 83(6):858–76.

[9] Lee, Y., E. D. Scheirer, and J.-W. Yang. Synthetic and SNHC audio in MPEG-4. *Signal Processing: Image Communication* (2000), 15(4-5):445–61.

[10] Girod, B., E. Steinbach, and N. Färber. Comparison of the H.263 and H.261 video compression standards. In *Standards and Common Interfaces for Video*, SPIE (Oct. 1995), CR60:233–51.

[11] Hartman, J., and J. Wernecke. *The VRML Handbook*. Reading, MA: Addison Wesley, 1996.

[12] Haskell, B. G., et al. Image and video coding: Emerging standards and beyond. *IEEE Trans. Circuits Syst. for Video Technology* (Nov. 1998), 8(7):814–37.

[13] Haskell, B. G., A. Puri, and A. N. Netravali. *Digital Video: An Introduction to MPEG-2*. New York: Chapman & Hall, 1997.

[14] Herpel, C., A. Elftheriadis, and G. Franceschini. MPEG-4 systems: Elementary stream management and delivery. In A. Puri and T. Chen, eds., *Multimedia Systems, Standards, and Networks*. New York: Marcel Dekker, 2000, 367–405.

[15] ISO/IEC. IS 10918-1: Information technology—digital compression and coding of continuous-tone still images: Requirements and guidelines. 1990. (JPEG).

[16] ISO/IEC. IS 11172: Information technology—coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbit/s. 1993. (MPEG-1).

[17] ISO/IEC. IS 13818-2: Information technology—generic coding of moving pictures and associated audio information: Video. 1995. (MPEG-2 Video).

[18] ISO/IEC. IS 13818-3: Information technology—generic coding of moving pictures and associated audio information—part 3: Audio, 1995. (MPEG-2 Audio).

[19] ISO/IEC. IS 13818-1: Information technology—generic coding of moving pictures and associated audio information: Systems, 1995. (MPEG-2 Systems).

[20] ISO/IEC. IS 13818-3: Information technology—generic coding of moving pictures and associated audio information—part 7: Advanced audio coding (AAC), 1997. (MPEG-2 AAC).

[21] ISO/IEC. IS 14772-1: Information technology—computer graphics and image processing—the virtual reality modeling language—part 1: Functional specification and UTF-8 encoding, 1997. (VRML).

[22] ISO/IEC. IS 14496-X: Information technology—coding of audio-visual objects, 1999. (MPEG-4).

[23] ISO/IEC. IS 14496-1: Information technology—coding of audio-visual objects—part 1: Systems, 1999. (MPEG-4 Systems).

[24] ISO/IEC. IS 14496-2: Information technology—coding of audio-visual objects—part 2: Visual, 1999. (MPEG-4 Video).

[25] ISO/IEC. IS 16500: Information technology—generic digital audio-visual systems, 1999. (DAVIC).

[26] ISO/IEC. Report of the formal verification tests on advanced coding efficiency ACE (formerly Main Plus) profile in version 2. Public document, ISO/IEC JTC 1/SC 29/WG 11 N2824, July 1999.

[27] ISO/IEC. Report of the formal verification tests on MPEG-4 coding efficiency for low and medium bit rates. Public document, ISO/IEC JTC 1/SC 29/WG 11 N2826, July 1999.

[28] ISO/IEC. CD 15938-3: MPEG-7 multimedia content description interface—part 3: Visual. Public document, ISO/IEC JTC1/SC29/WG11 W3703, La Baule, France, October 2000.

[29] ISO/IEC. CD 15938-5: Information technology—multimedia content description interface—part 5: Multimedia description schemes. Public document, ISO/IEC JTC1/SC29/WG11 N3705, La Baule, France, October 2000.

[30] ISO/IEC. IS 14496-3: Information technology—coding of audio-visual objects—part 3: Audio, 2000. (MPEG-4 Audio).

[31] ISO/IEC. IS 14496-6: Information technology—coding of audio-visual objects—part 6: Delivery multimedia integration framework (DMIF). 2000. (MPEG-4 DMIF).

[32] ISO/IEC. Overview of the MPEG-7 standard (version 4.0). Public document, ISO/IEC JTC1/SC29/WG11 N3752, La Baule, France, October 2000.

[33] ISO/IEC. MPEG-7 visual part of experimentation model version 9.0. Public document, ISO/IEC JTC1/SC29/WG11 N3914, Pisa, January 2001.

[34] ITU-R. BT. 500-10: Methodology for the subjective assessment of the quality of television pictures, 2000.

[35] ITU-R. BT.601-5: Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios, 1998. (Formerly CCIR601).

[36] ITU-T. Recommendation G.711: Pulse code modulation (PCM) of voice frequencies, 1988.

[37] ITU-T. Recommendation G.722: 7 kHz audio-coding within 64 kbit/s, 1988.

[38] ITU-T. Recommendation G.723.1: Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s, 1988.

[39] ITU-T. Recommendation G.728: Coding of speech at 16 kbit/s using low-delay code excited linear prediction, 1992.

[40] ITU-T. Recommendation T.81—Information technology—Digital compression and coding of continuous-tone still images—Requirements and guidelines, 1992. (JPEG).

[41] ITU-T. Recommendation H.261: Video codec for audiovisual services at $p \times 64$ kbits, 1993.

[42] ITU-T. Recommendation V.34: A modem operating at data signaling rates of up to 28800 bit/s for use on the general switched telephone network and on leased point-to-point 2-wire telephone-type circuits, 1994.

[43] ITU-T. Recommendation H.262: Information technology—generic coding of moving pictures and associated audio information: Video, 1995.

[44] ITU-T. Recommendation H.324: Terminal for low bit rate multimedia communication, 1995.

[45] ITU-T. Recommendation H.223: Multiplexing protocol for low bit rate multimedia communication, 1996.

[46] ITU-T. Recommendation H.320: Narrow-band visual telephone systems and terminal equipment, 1997.

[47] ITU-T. Recommendation V.25ter: Serial asynchronous automatic dialling and control, 1997.

[48] ITU-T. Recommendation H.225.0: Call signaling protocols and media stream packetization for packet based multimedia communications systems, 1998.

[49] ITU-T. Recommendation H.245: Control protocol for multimedia communication, 1998.

[50] ITU-T. Recommendation H.263: Video coding for low bit rate communication, 1998.

[51] ITU-T. Recommendation H.323: Packet-based multimedia communications systems, 1998.

[52] ITU-T. Recommendation V.8 bis: Procedures for the identification and selection of common modes of operation between data circuit-terminating equipments (DCEs) and between data terminal equipments (DTEs) over the public switched telephone network and on leased point-to-point telephone-type circuits, 1998.

[53] ITU-T. Recommendation V.8: Procedures for starting sessions of data transmission over the public switched telephone network, 1998.

[54] Koenen, R., Profiles and levels in MPEG-4: Approach and overview. *Signal Processing: Image Communications* (2000), 15(4-5):463–78.

[55] Mitchell, J. L., W. B. Pennebaker, C. E. Fogg, and D. J. LeGall. *MPEG video compression standard.* Digital Multimedia Standards Series. Bonn; Germany: Chapman and Hall, 1996.

[56] Mokhtarian, F. S., S. Abbasi, and J. Kittler. Robust and efficient shape indexing through curvature scale space. In *British Machine Vision Conference*, 53–62, Edinburgh, England, 1996.

[57] Musmann, H. G., and J. Klie. TV transmission using a 64 kbit/s transmission rate. In *International Conference on Communications*, Boston, MA, IEEE 23.3.1–5, 1979.

[58] Okubo, S., Reference model methodology—a tool for the collaborative creation of video coding standards. *IEEE* (Feb. 1995), 83(2):139–50.

[59] Orchard, M. T., and G. J. Sullivan. Overlapped block motion compensation: An estimation-theoretic approach. *IEEE Trans. Image Process* (1994), 3:693–99.

[60] Ostermann, J., and A. Puri. Natural and synthetic video in MPEG-4. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'98)*, IEEE, Seattle, WA, USA (Nov. 1998), 3805–09.

[61] Puri, A. Video coding using the MPEG-2 compression standard. *SPIE Visual Communications and Image Processing* (Nov. 1993), 1199:1701–13.

[62] Puri, A., and A. Wong. Spatial domain resolution scalable video coding. *SPIE Visual Communications and Image Processing* (Nov. 1993), 1199:718–29.

[63] Puri, A., L. Yan, and B. G. Haskell. Temporal resolution scalable video coding. *International Conference on Image Processing (ICIP 94)*, IEEE, Austin, Texas, USA (Nov. 1994), 2:947–51.

[64] Salembier, P., and J. R. Smith. MPEG-7 multimedia description schemes. *IEEE Trans. Circuits Syst. for Video Technology* (June 2001), 6:748–759.

[65] Schulzrinne, H., S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications. IETF RFC 1889, Jan. 1996 (Available via ftp://ftp.isi.edu/in-notes/rfc1889.txt).

[66] Signés, J., Y. Fisher, and A. Eleftheriadis. MPEG-4's binary format for scene description. *Signal Processing: Image Communications* (2000), 15(4-5):321–45.

[67] Signés, J., Y. Fisher, and A. Elftheriadis. MPEG-4: Scene representation and interactivity. In A. Puri and T. Chen, eds., *Multimedia Systems, Standards, and Networks*. New York: Marcel Dekker, 2000:407–47.

[68] Tekalp, A. M., and J. Ostermann. Face and 2-D mesh animation in MPEG-4. *Signal Processing: Image Communications, Special Issue on MPEG-4* (Jan. 2000), 15:387–421.

[69] Verne, J. In the twenty-ninth century. The day of an American journalist in 2889. In *Yesterday and Tomorrow*, London: Arco, 1965. (Translation by I. O. Evans of *La Journée dún journaliste américain ex 2890*, Amiens: Atelier deGué, 1979, a short story written in 1888 and first published in English in 1889 as *In the Year 2089.*)

[70] Wiegand, T., M. Lightstone, D. Mukherjee, T. Campell, and S. K. Mitra. Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard. *IEEE Trans. Circuits Syst. for Video Technology* (April 1996), 6(2):182–90.

第 14 章　视频通信中的差错控制

在前几章中，我们已经讨论了各种视频编码技术和标准。对于高效视频通信，降低原始视频码率只是必要的步骤之一。另一个同等重要的任务是要处理通信网络中的差错和损失。数据通信一般不会受到严格的延迟约束，因此能够采用基于重传机制的网络协议以确保无误差传送。实时视频则不同，它对延时敏感，因此不容易利用重传的方法。视频编码中广泛使用预测编码和可变长编码，致使压缩的视频特别容易受到传输差错的损害，在存在误差情况下，成功的视频通信要求仔细地设计编码器、解码器以及其他系统层。

在本章中，我们给出目前已有的用于视频通信中差错控制的方法。我们首先说明视频通信差错控制的必要性和所面临的挑战，并概要介绍已有的各种方法(14.1 节)。为了有助于理解视频通信中的差错控制问题，在 14.2 节中描述各种视频业务的 QoS 要求和不同网络的特性。14.3 节给出传输层的差错控制机制。14.4 和 14.5 节对差错复原编码和错误隐藏技术进行了综述。14.6 节描述基于编码器和解码器交互的技术。最后，14.7 节总结了 H.263 和 MPEG-4 标准所采用的差错复原工具。

14.1　动机和方法概述

典型的视频通信系统包括 5 个步骤，如图 14.1 所示。视频编码器首先对视频进行压缩以降低码率，压缩比特流随后被分割成固定长度的或可变长度的包，并与其他类型的数据(例如音频)复用。如果网络保证无比特差错传输，则可直接把包传送到网络上。否则，这些数据包一般要经过信道编码，通常使用前向纠错(FEC)和交织，以进行误码保护。在接收端，将接收到的包进行信道解码和解包，得到的比特流随后输入到视频解码器以重建原始视频。实际上，许多应用系统把打包和信道编码嵌入在信源编码器作为网络的一个适配层。

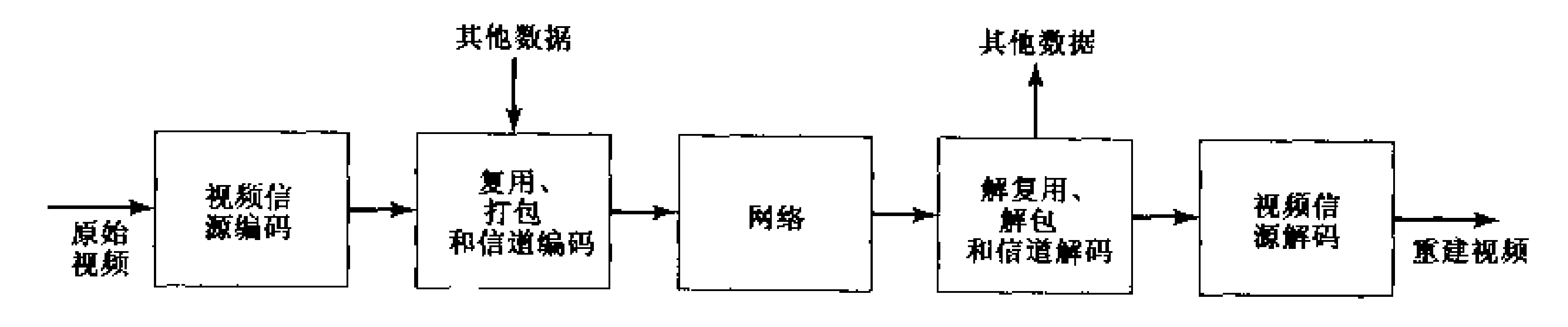


图 14.1　一个典型的视频通信系统

传输错误可粗略地分为两类：随机比特错误和删除错误。随机比特错误是由物理信道的不完善引起的，它导致比特反转、插入和删除。随机比特错误的影响取决于编码方法和受影响的信息内容，从可忽略到不可忍受。当采用固定长度编码时，随机比特错误将只影响一个码字，引起的损害一般是可接受的。但如果采用可变长编码(VLC，例如哈夫曼编码)，随机比特错误可能使编码信息失去同步，从而导致许多后续比特不可解码，直到下一个同步码字出现为

止。另一方面，删除错误可能由包交换网络(如因特网)中的包丢失、存储媒体中由于物理缺陷导致的突发性差错、或者短时间的系统故障所引起。VLC 编码流中的随机比特差错也可引起删除错误，由于单个比特差错可导致许多后续比特不可解码，从而使之不可用。由于连续的比特段被丢失或破坏，删除错误(包括那些由随机比特错误引起的删除)的影响比随机比特错误更具有破坏性。

由于以下几个原因，视频通信中的差错控制具有相当的挑战性。首先，由于信源编码器使用时间预测编码和 VLC，压缩的视频流极容易受到传输差错的破坏。由于使用时间预测，一个错误地恢复的样点可能导致同一帧和后续帧中后续样点发生错误，如图 14.2 所示。注意，不只在时间上产生误差积累，也会由于基于运动补偿的预测在空间上产生误差积累。其次，由于使用 VLC，比特错误的影响等价于删除错误的影响，引起视频帧中大面积的破坏。一般，在基于块的混合编码器中(9.3.1 节)，每一个块组(一般是宏块行)是由重同步标志打头的，它帮助解码器重新同步。如果在 GOB(块组)的中间检测出传输误码，即丢弃整个 GOB。解码器在下一个 GOB 再重新开始解码。为了说明由传输误码引起的视觉效应，图 14.3 示出了一个出现包丢失时重建帧的典型例子。在这个例子中，损坏的 GOB 用基于运动补偿的时间内插方法(14.5.1 节)进行恢复。当丢包率不太高时(图中丢包率为 3%和 5%的情况)，这种方法能在某种程度上隐藏错误，但当有太多的包丢失时(图中 10%的丢失情况)就会彻底破坏比特流。注意，所示帧中的人为失真不只是由于本帧而且还由于前面帧的数据包丢失所引起。

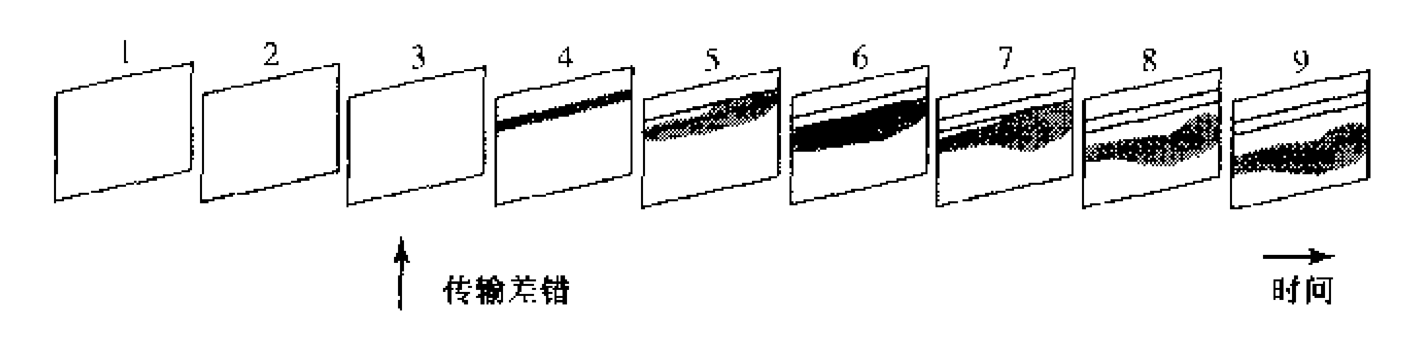


图 14.2 时空误差积累图示

为了使压缩比特流对于传输误码能够复原，必须在流中加入冗余度，以使其能够检测和纠正错误。这一般在信道用 FEC 实现，对信源编码器产生的比特流进行前向纠错编码。经典的香农信息论指出，可以分别设计信源和信道编码器，只要用低于信道容量的速率表示信源，就可实现压缩比特流的无差错传输。因此，对于指定的失真，信源编码器应该尽可能地压缩信源(低于信道容量)，信道编码器可以通过 FEC 将冗余度加入到压缩码流中，以便能够纠正传输误码。然而，这种理想的无差错传送只有当实现 FEC 时具有无限延迟才能达到，这在实际中是不能接受的。因此，信源和信道联合编码经常是更可行的方案，它在信源编码和信道编码之间分配总的冗余量。所有的差错复原编码方法实质上都是在这个前提下工作的，有意使信源编码器的效率比可能达到的低，以使压缩流中出现差错或丢失的比特对重建视频质量不会有严重的影响。通常这是由仔细地设计预测编码环路和可变长度编码器以限制误差积累的程度实现的。

当由于传输差错而丢失一个图像样点或样点块时，解码器可利用时域和空域相邻样点的内在相关性，根据相邻的已接收样点估计它们。这种技术称为错误隐藏。由于实际的信源在编码过程中没有完全去除信号的冗余度，所以这是可能的。与差错复原信源编码不同，错误隐

藏的优点是不利用任何额外的比特率[1]，但在解码器中要增加计算的复杂度。

最后，为了使在信源编码器中嵌入的冗余有用以及便于在解码器中进行错误隐藏，编解码器和网络传输协议必须协调一致。例如，如果比特流中某些比特比其他比特更重要，那么应该给重要的部分分配一组更严格的 QoS 参数以便在网络上传输。为了抑制误差积累，网络也可提供反馈信道，以使编码器知道解码端重建信号的哪部分受到了破坏，从而不用这部分来预测将来的样点。

总之，为视频传输而设计的差错控制机制可分为 4 类：(1)应用于传输层，包括信道编码器、打包器/复用器以及传输协议；(2)在信源编码器引入，使比特流对潜在差错更具差错复原性；(3)在解码器通过错误检测，以隐藏差错的影响；(4)需要信源编码器与解码器之间交互作用，使得发送端能够根据在解码端检测到的丢失状况修改其操作。我们将在 14.3～14.6 节中分别描述这 4 类技术。



图 14.3　传输错误对采用 H.263 标准的压缩视频流的影响。左上图：没有传输错误，图像质量可达比特率所限。右上图：3%的包丢失；左下图：5%的包丢失；右下图：10%的包丢失

14.2　典型的视频应用系统和通信网络

在本节中，我们说明不同类型的视频应用系统和实际网络的特性。这些是要考虑的重要因素，因为差错控制的必要性和技术的有效性取决于应用类型以及底层信道特性和网络协议。

14.2.1　视频应用系统的分类

当考虑视频传输中的差错控制时，重要的是知道应用系统是否要求实时传送，以及允许的最大平均端到端的延迟（称为等待时间）和延迟变化（称为抖动）是多少。“实时”传送

[1] 为了便于差错隐藏，通常要在编码器中以约束预测和数据交织的形式插入少量的冗余。

意味着压缩数据是以与编码视频信源的速率相匹配的速度传送。例如，如果源视频是以10 Mbps编码的，那么要传送此信源的通信信道的吞吐量[①] 至少应该是 10 Mbps。注意，可以实时（如在视频会议应用系统中）或离线（如在视频流应用系统中）产生源视频信号。除了简单的下载外，所有的视频应用系统都要求实时传送。但是某些应用系统（例如流媒体）可以允许相对较大的播放延迟。在这种情况下，接收端可用一个大的缓冲区来平滑抖动，使得解码视频在初始的播放延迟之后，能够以一个恒定的帧率播放。对于这些应用系统，可用有限数目的重传来处理传输差错。下面，我们根据视频应用系统在实时传送、等待时间和抖动方面的要求对它们进行分类。

注意，由视频信源到信宿的端到端延迟可能与几个因素有关（见图 14.4）：编码器处理延迟 ΔT_e（包括采集数据和编码）；编码器缓冲区延迟 ΔT_{eb}（为了平滑压缩比特流中的速率变化），传输延迟 ΔT_c（一般由传输本身引起的很小的延迟，以及由于排队和在基于包的网络中可能重传而引起的延迟），解码器缓冲区延迟 ΔT_{db}（为了平滑传输抖动）以及解码器处理延迟 ΔT_d（包括解码和为恒定帧播放的显示缓冲区）。当然，若信源是预先压缩的，如在视频广播和流媒体应用系统中，则不考虑编码延迟。在本节中，我们主要讨论传输延迟，假设由视频编解码以及编解码缓冲区引起的延迟对应用系统来说是可接受的，并且这些延迟是相对固定的。

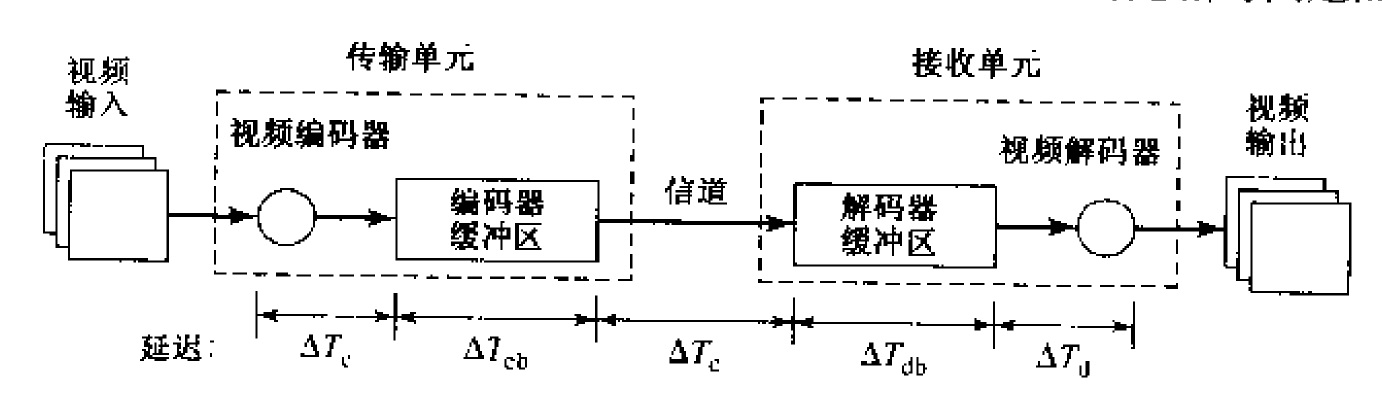


图 14.4 在视频通信系统中引起端到端延迟的因素

交互式双向视频通信 这类例子包括电视会议、视频电话、虚拟教室等。这类应用系统有非常严格的延迟要求。为有效地通信，等待时间和抖动必须保持在一定的限度内。例如对于洲际电话会谈，ITU-T G.114 标准建议可接受质量的单向最大延迟是 150 ms。假如网络管理者意识到延迟对用户应用系统的影响，150～400 ms 是可接受的[37]。同样的限定也适用于视频会议应用系统。400 ms 以上的延迟对于有效的通信系统是不可忍受的。同样，当播放音频和视频时，为保持声像同步，应限制视频相对于音频的延迟。通常，在发送端和接收端中有意稍微延迟音频，以保持声像同步。

对于这类应用系统，必须实时实现编码和解码。否则，到来的帧可能会堆积在编码器中，或者在解码器中不得不丢弃所接收的比特。为了满足严格的延迟要求，编码器和解码器的缓冲区一般很小。重传一般是不可接受的。网络中的抖动也必须受限。由于有这些限制，交互式应用系统被认为是在各种类型的视频通信中要求最苛刻的。

幸运的是，在空间和时间分辨率方面只需要达到低到中等的视频质量。例如，5～10 fps

① "吞吐量"是指有效的端到端传输速率。例如，一个网络可能具有 10 Mbps 的原始带宽，但某些数据由于丢包或超时需要重传，使得实际吞吐量低于 10 Mbps。

的 QCIF 对于视频电话应用系统是可接受的[①]。而 10 ~ 20 fps 的 CIF 对于大多数视频会议足以令人满意。而且,中等程度的压缩和传输人为失真通常是可容忍的。

单向视频流 在这类应用系统中,现场或预先压缩的视频信源被传送到一个或多个信宿。接收端不必等待整个视频文件到来,在可能最多几秒的初始播放延迟后,就开始解码并播放视频。根据接收者的数目,应用系统可进一步分为广播、组播或单播。电视广播是最广泛采用的视频应用系统。视频组播,例如 Mbone(一种多应用系统,它允许任何因特网上的用户观看会议或讲座的实况视频)是另一种重要的应用。正日益流行的一种应用系统是从预先压缩的视频数据库通过因特网播放流视频,一般由单个用户接收。

显然,带宽需求取决于视频资料的类型。对于在地面、有线或直播卫星 (DBS)信道上的数字电视广播,视频节目一般为 BT.601 分辨率,用 MPEG-2 格式压缩,具有 3 ~ 10 Mbps 的比特率。对于 HDTV,比特率可高达 20 Mbps。对于因特网或无线网络上的组播和单播,由于每个用户的带宽有限以及带宽和延迟可变,一般使用较低质量的视频(CIF 或以下,用 MPEG-1 或 MPEG-4 或其他专有格式压缩)。

由于发送端和接收端之间没有交互式通信,所以等待时间可能相当长,但必须限制抖动,以使接收到的流能被解码,并且在初始延迟后以恒定的帧率平滑地播放。电视广播使用带宽有保证且抖动非常低的专用信道。对于在因特网上的视频组播或单播,由于网络状况的可变性,抖动可能很大,在接收端一般都设置一个大的平滑缓冲区,以减小抖动。缓冲区越大,播放延迟越长,但显示的视频越平滑。

在因特网的组播或单播情况下,潜在接收者可用不同的接入线路(从 100 Mbps 的快速以太网到低速的无线调制解调器线路)连接到网上。另外,接收端可能具有非常不同的计算能力(由强大的工作站到电池供电的手持设备)。视频服务器必须考虑这些不定因素。例如,采用可分级编码(第 11 章),具有不同带宽和计算能力的用户可选择抽取压缩视频流的不同部分。在第 15 章将更详细地讨论因特网和无线 IP 网络上的流视频。

显然,现场信源与预先压缩信源的主要区别在于现场信源必须实时压缩,这就显著增加了传输端系统的复杂度。编码处理和编码器缓冲区也将引起额外延迟。

单向视频下载 在这种情况下,预先记录的视频(和音频) 被下载到信宿。直到整个视频到达之后,接收端才开始播放资料。这种应用系统在延迟方面的需求最低,尽管过大延迟(由于非常低的传输带宽引起)可能造成接收端在通信会话中途中断下载。可分级编码和累进传输是所希望的,以便相对快地传送一个低分辨率的视频版本。由于宽松的延迟要求,所以可采用重传来处理数据的任何丢失或错误传送部分。除了一般用于数据传送的差错控制机制外,不需要特殊的差错控制方法。

14.2.2 通信网络

在本小节中,我们描述不同类型的网络和可能用于视频传输的有关协议。对于每种网络,我们说明它在带宽、延迟、抖动和丢失率(包括比特错误和丢包)方面的特性以及在这些网络上实现的典型视频应用。

公共交换电话网络(PSNT) PSNT 指简单的老式电话系统(POTS)。PSTN 的最大优点是

① 在这样低的帧率下,视频来回旅程的延迟经常超过前面引用的 400 ms 极限,会引起声像不同步的问题。

它几乎可以被美国或(从更大范围来讲)世界上的任何人接入。使用最先进的调制解调技术,目前最快的传输速率已达到 56 kbps,但这对于传送具有满意质量的视频仍然太低。为了在 PSTN 上进行多媒体通信,开发了 ITU-T H.324 标准系列(13.3.2 节)[36]。在基于 H.324 的系统中,通过调制解调器控制机制来调节误码率与调制解调线路比特率之间的折中。大多数系统采用强大的差错控制编码,以一些比特率为代价达到非常低的差错率[81]。

即使在 56 kbps 下(一般,56 K 调制解调器上可达到的有效载荷传输率比这低得多),也只能提供一个小窗口视频(QCIF 或更小)。尽管这对视频通信的目的可能够用,但远不能达到令人满意的程度。用 ADSL(非对称数字用户环路)技术[54]可实现更高的速率,在下行链路方向(由办公中心到家)上高达 6 Mbps。ADSL 是视频点播业务中播放 MPEG-1 和 MPEG-2 流的主要传输媒体之一。涉及 ADSL 上多媒体传输问题的详细讨论参见参考文献[96]。

综合业务数字网(ISDN) ISDN 是第一个使用数字传输的公共网络[66]。码率可以是 64 kbps 基本信道(称为 B 信道)的倍数,乘数因子 p 的范围从 1 至 24,总计带宽相当于 64~1536 kbps。ISDN 的基本速率接口为 2B+D 信道,D 信道是 16 kbps,它用于返回信令。以 128 kbps($p=2$)传送信号只能达到很低质量的视频(例如 10 fps 的 QCIF)。为获得更好的视频质量(即 15~30 fps 的 CIF),至少需要 384 kbps($p=6$)。

与电话线路一样,ISDN 上的通信是基于电路交换的。也就是说,一旦建立连接,ISDN 连接就为特定的会话所专用。由于这个原因,连接是非常可靠的。第一个用于视听会议的ITU-T 建议系列 H.320 是为 ISDN 而开发[40]。目前绝大多数视频会议和视频电话设备采用 H.320。H.320系统所用的多路复用协议 H.221[39]提供了一个具有固定视频比特率的面向比特的、实际上无差错的视频传输信道。除了视频编码标准强制的帧内宏块刷新机制外,不需要其他差错控制工具[81]。

宽带 ISDN(B-ISDN) 刚才所述的具有 64 kbps 基本信道速率的 ISDN 业务,应更精确地称为窄带 ISDN 或 N-ISDN。在现有的双绞线局域环路布线上可提供这种业务。宽带 ISDN 或 B-ISDN[66]是指提供高宽带信道的 ISDN 业务,包括具有 384 kbps 速率的 H0 信道,具有 1536 Mbps速率的 H11 信道以及具有 1920 Mbps 速率的 H12 信道。这需要使用更高带宽的同轴电缆或光纤。B-ISDN 的带宽也规定为 64 kbps 的倍数,乘数因子范围从 1 至 65 535。

为有效地利用带宽,B-ISDN 使用具有固定尺寸包(称为单元)的 ATM 包交换技术。由于没有为所建立的连接建立专用线路,因此它不如电路交换 ISDN 可靠(即具有较大的延迟变化和潜在单元丢失)。很短的单元(53 字节,其中 48 字节是有效载荷)使它适合于低延迟要求的实时应用系统。在这类网络中,可能因线路拥塞引起单元丢失,尽管丢失率相当低;对于视频业务[3],典型的单元丢失率(CLR)范围是 $10^{-6}\sim10^{-4}$。一般,单元丢失可以被认为是包丢失的一种形式,由此可以认为单元是极小的包。然而,处理单元丢失的机制不同,因为从差错复原与开销的折中角度讲,在每个单元的开始处加同步标志效率不高。

ATM 网络提供四种类型的业务:恒定比特率(CBR)、可变比特率(VBR)、可用比特率(ABR)以及未指定比特率(UBR)。通常,视频以 VBR 业务传送,压缩比特流必须满足平均速率和峰值速率方面的预置条件。允许速率变化令其可以选择视频编码参数来保持相对恒定的视频质量,这是所期望的特性。ATM 网络的详细综述以及与在 ATM 上视频传输有关的问题参见参考文献[3, 5]。

因特网 因特网是由称为路由器的计算机互联的多个物理网络所组成。最基本的组块称

为局域网(LAN),其带宽一般为 10 ~ 100 Mbps。在本地区域相互连接的 LAN 构成城域网(MAN),最后,几个相互连接的 MAN 形成广域网(WAN)。因特网上的通信是基于包交换的,也就是说,把所要传送的数据分成包,每个包都给定一个包头来指定信源和信宿的地址。每个包是独立传送的。在中间网络节点处(例如交换机或路由器)可能由于缓冲区溢出而丢弃包,或由于过长的排队延迟而被认为丢包。对于实时视频应用系统,任何在允许延迟时间之后到达的包也被认为丢失。包的传输是由 TCP/IP 协议栈控制的,它包括三个重要的协议:因特网协议(IP),用于不可靠的无连接包传输业务的用户数据报协议(UDP),以及用于可靠的流服务的传输控制协议(TCP)。IP 是低层协议,UDP 和 TCP 在它的上面工作。TCP 通过使用自动重传请求(ARQ)来保证传送[16]。

因特网及其有关的 TCP/IP 协议最初是为对延迟不敏感的数据通信设计的。对于实时业务,例如视频会议和流视频,TCP 中的重传机制并不适合。为了支持这类应用,开发了实时传输协议(RTP)和与其相配的实时控制协议 (RTCP)。RTP 构建于 UDP 之上,但在传输头中增加了信息,包括序列号以及定时和同步信息。RTP 允许在接收端根据序列号检测丢失的包以及丢弃迟到的包。RTCP 是为向 RTP 话路的参与者提供 QoS 反馈(例如丢包率)设计的,以便发送端能相应地调整它的工作(例如调整传送率或改变差错控制机制)。有关支持连续媒体业务的因特网协议,包括 RTP/RTCP、媒体点播的实时流协议(RTSP)、因特网电话的话路启动协议(SIP)、用于广播应用系统的话路通告协议(SAP)以及话路描述协议(SDP)的细节参见参考文献[62]。

通常,加在不同网络层的传输头会给一个包增加大约 40 字节。为了获得有效载荷与开销之间的合理折中,通常采用大约 1500 字节的大数据包。通过使用检验和,包中的某些比特差错可以被检测出来。如果检测到任何比特差错,则这个包就被完全丢弃掉。因此任何收到的包都可认为是无比特差错的。丢包率依赖于网络状况,在高度防备的私人 IP 网络情况下可能是 0%,对于在因特网峰值期间的远距离连接,则可高达 30% 以上[6,81]。使用 RTP 时,用序列号可以很容易地发现丢失的包。此信息可传送到视频解码器,启动解码器端的错误隐藏。

最广泛使用的支持本地因特网通信的物理网络是以太网,具有 10 Mbps 的带宽。更先进的网络包括快速以太网和光纤分布数据接口(FDDI)网络;二者都支持高达 100 Mbps 的码率。这种较快的网络通常用于企业内部网环境中。

由于因特网无处不在,它已经被预见为传送各种视频业务的未来平台。然而,由于因特网是一个最尽力的网络,它不能保证无差错或及时传输,所以因特网上视频通信的差错控制是一个具有挑战性的研究领域。关于因特网和 RTP/RTCP 协议的更深入讨论参见第 15 章,其中主要讨论因特网上的视频流应用系统。

无线网络　针对不同的环境和应用系统,存在各种各样的无线网络。这些系统具有非常不同的容量、覆盖范围和误码特性。在下文,我们简要地概括不同的无线业务。我们希望指出的是,无线通信是一个发展迅速的技术,在这本书到达读者手中时,更新、更可靠和更快的业务很可能已开始实用了。

蜂窝网络[27]传统上是为移动语音通信业务开发的;但现在已经发展到允许数据通过无线调制解调器传送。码率限制到 20 kbps 以下(有效载荷码率更低)。蜂窝网络也可以连接到基于包交换的无线数据网络,允许在移动用户与因特网上的有线终端之间传输数据(例如万维网浏览应用)。例如,在全球移动通信系统(GSM)中使用的基本 GPRS(通用包无线电业务) 无

线接入网络,利用时分多址(TDMA),在单个时间段内可提供 9~21.4 kbps 的有效载荷比特率;而改进的 GPRS 无线接入技术(称为增强数据率的 GSM 进化或 EDGE)可提供 8.8~59.2 kbps 的比特率[9]。通过使用多个时间段,原始码率可高达 170 kbps。可用码率依赖于位置,用户离基站越近,码率越高。即将到来的第三代(3G)无线系统声称,通过用更高的载频、更宽的带宽、更复杂的多路接入、差错控制以及信号检测技术,可提供更高的码率和无处不在的覆盖范围。户外码率范围将从 144 到 384 kbps,取决于用户的运动速度以及终端之间的距离;室内码率将至少是 2 Mbps。数据传输的误码率(BER)低于 10^{-6}。实时视听通信将是 3G 业务的一部分。通常,任何无线信道都有相当大的噪声,具有很高的 BER。但是使用速率自适应调整、FEC 和 ARQ 会得到一个几乎无差错的数据传输环境。对于 ARQ 必须受限的视频传输,我们必须处理相当高的 BER 和丢包率。这将在 14.3.2 节中进一步讨论,我们考虑用 H.223 多路复用协议进行无线视频传输。

无线局域网指用无线连接互连成局域网的一组信息设备(通常在室内,静止或运动缓慢)。可使用 IP 协议或 ATM 协议,分别称为移动 IP 和无线 ATM。传统 IP 和 ATM 协议需适应无线环境的主要挑战是,当移动终端从一个接入点的覆盖区切换到另一个覆盖区时,应为移动节点提供连续的网络连接,并且处理区域交接。可达到的比特率取决于载频。例如,遵从 IEEE 802.11 标准的局域网可以具有的比特率高达 11 Mbps(有效负载率低一些)。当用户靠近接入点时,无线局域网中的连接是很好的(例如 BER 低于 10^{-5});而当用户远离接入点时,则很差(例如 BER 高于 10^{-2})。丢包率取决于包的尺寸以及包内所含的差错检测和纠正方法。

宽带无线 IP 网是指家庭或商业用户用微波无线电通过小的屋顶天线与基站通信,接入因特网。包括多信道多点分布业务(MMDS)和本地多点分布业务(LMDS)。LMDS 采用较高的载频,因此允许比 MMDS(高达 1 Mbps)更高的码率(一般达到 100 Mbps,也可高达 600 Mbps)。然而,LMDS 只在基站与客户站点之间很短的范围内有效(是 3 英里和 30 英里的对比关系)。

表 14.1 概括了我们已经讨论过的各种网络的特性。各种无线网络的综述可参见参考文献[9]。

表 14.1 不同类型的无线网络

网络	码率	移动性	范围	信道质量
蜂窝网	低(≤20 kbps)	高(车载)	长(2500 米)	差
无线数据网	中(64~384 kbps)	高(车载)	长(2500 米)	差
无线局域网	高(2~25 Mbps)	低(室内)	短(50 米)	依赖于位置
无线 IP 网	高(1~600 Mbps,下行链接)	低(室内)	长(3~30 英里)	好

在无线网络上传输视频的困难在于低带宽和高误码率(随机误码以及由多径衰落效应导致的长突发性差错),而最重要的是可用带宽和差错特性的波动。当涉及到移动站时,这种变化尤其剧烈。因此,无线视频通信系统在速率和差错复原两方面都必须具有自适应性。为了满足这种要求,对于无线视频传输的大多数建议都采用具有不平等差错保护的分层编码。值得注意的是,对于实际应用系统,用相当复杂的 FEC 码来大幅减少 BER。实际系统一般都在信道编码后采用某种形式的信道复用器,它也经常包含传输协议的功能,由此进一步降低了误码率。

大多数无线交互式多媒体通信系统都采用 H.223[38]及其"移动扩展"作为面向比特信道上的传输/复用协议[81]。移动扩展形成了一个五级的分层结构,它可实现复用器本身的抗误码健壮性

与复用器开销之间的可分级的折中。在平均突发差错长度为相邻 16 比特和误码率为 10^{-5} 与 10^{-3} 之间的情形下,对 H.223 上层的视频编码进行了大量研究。H.223 以可变尺寸包的形式传送包括压缩视频在内的媒体数据。一般包的尺寸大约为 100 字节,以保证好的延迟特性。如果误码对 H.223 协议结构的破坏超出了所采用级的修复范围(一种称为复用差错的情况),那么整个包也可能丢失。因此,在 H.223 上层的视频传输必须解决丢包问题。

关于无线系统视频传输中所涉及问题的讨论见参考文献[76,25]。无线 IP 网络上的流视频将在第 15 章中进一步讨论。

广播信道:地面、电缆、卫星 这种传输环境用于采用 MPEG-2 视频编码和传输流的数字电视广播(包括 HDTV 节目)。压缩数据通过固定长度(188 字节)的传输包进行传输。不管是有线还是无线物理层,信道编码器和 MPEG-2 传输层确保良好大气条件下几乎无差错的环境。对于 SDTV,可用带宽被划分为节目频道,每个节目频道的传输带宽在 3 ~ 10 Mbps 之间。对于 HDTV 一般分配 20 Mbps 的带宽。有关使用直接广播卫星(DBS)的数字电视业务,请参看参考文献[55]。

表 14.2 概括了主要视频通信应用系统的特性,包括所用标准、目标网络以及传输特性(例如包的尺寸和典型的误码率)。

表 14.2 主要视频通信应用系统的特性

应用和标准系列	复用协议	视频编码标准	典型视频比特率	包尺寸	差错特性
ISDN 视频电话(H.320)	H.221	H.261, H.263	64 ~ 384 kbps	N/A	实际上无错($BER = 10^{-10} \sim 10^{-8}$)
PSTN 视频电话(H.324)	H.223	H.263	20 kbps	100 字节	很少的比特错误和丢包
移动视频电话(H.324 无线)	具有移动扩展的 H.223	H.263	10 ~ 300 kbps	100 字节	$BER = 10^{-5} \sim 10^{-3}$ 偶尔丢包
传输包网络上的视频电话(H.323)	H.225/RTP/UDP/IP	H.261, H.263, H.262	10 ~ 1000 kbps	≤1 500 字节	BER = 0,0 ~ 30% 丢包
地面/有线/卫星电视	MPEG-2 系统	MPEG-2 视频	6 ~ 12 Mbps	188 字节	几乎无错,依赖于天气
"本地"ATM 上的视频会议(H.310, H.321)	H.222.0	H.262	1 ~ 12 Mbps	53 字节(ATM 单元)	几乎无错($CLR = 10^{-6} \sim 10^{-4}$)

N/A:不可应用;H.262 等价于 MPEG-2 视频,而 H.222.0 等价于 MPEG-2 系统。

14.3 传输层差错控制

从本节开始,我们讨论为视频通信开发的各种差错控制机制。我们从传输层差错控制开始,因为通常这是差错控制最重要的部分。它提供一个基本的 QoS 级,通过编码器和解码器中的其他差错控制机制还可以进一步提高。可以在信道编码器、打包器/复用器以及传输协议层进行传输层差错控制。在本节分别对这些方面进行讨论。

14.3.1 前向纠错(FEC)

FEC 是数据通信中用于检错和纠错的最常见方法[52]。然而,由于 FEC 增加了传输开销,

从而降低有效载荷数据的可用带宽，在带宽有限但可以容忍一定程度损失的视频业务中，必须谨慎使用。

当将 FEC 直接应用于压缩比特时，它只对比特错误占支配地位的信道有效。例如，在用于 ISDN 视频会议的 H.261 标准中[35]，计算一个 18 比特的纠错码并且附加到 493(一个填充比特，492 个编码比特)个视频比特上，形成 BCH(511,493)码。加入一个附加的成帧比特后，所得到的数据组成包含 512 比特的一帧。FEC 码能在每帧中纠正单个比特错误和检测出两个比特错误。相同的 FEC 机制也能用于 H.263[42,附件 H](14.7.1 节)。然而，当把 H.263 用于无线网或因特网上传输的视频时，由于突发性差错一般长于两个比特，这种方法不是很有用，因此很少使用。本节中讨论的其他传输层差错控制技术更为有效。

对于基于包交换的传输来说，应用纠错困难得多，因为当出现一个包丢失时，必须恢复好几百个比特。通常 FEC 跨越数据包使用，使得包丢失将只导致 FEC 块中的一个字节丢失。例如，在 Lee 方法[51]中，用里德－索罗蒙(Reed－Solomon，RS)编码与块交织相结合来恢复丢失的 ATM 单元。如图 14.5 所示，对每个 28 字节的数据块进行 RS(32,28,5)编码，形成一个 32 字节的块。在存储器中逐行应用 RS 编码直到第 47 行，然后从存储器中逐列读出数据形成 32 个 ATM 单元的有效载荷，并附加一个字节指明序列号。这样，在去交织后，在解码器中检测到丢失一个单元对应于每个 32 字节的行中擦除一个字节。32 个单元中丢失最多 2 个单元都能够恢复。大联盟 HDTV 广播系统已采用类似的技术来抵抗传输误码[10]。在参考文献[2]中 Ayanoglu 等研究了把 FEC 用于无线 ATM 中的 MPEG-2 视频。用 FEC 在字节级进行随机误码校正，在 ATM 单元级进行单元丢失恢复。这些 FEC 技术已在单层或双层 MPEG 数据中采用。

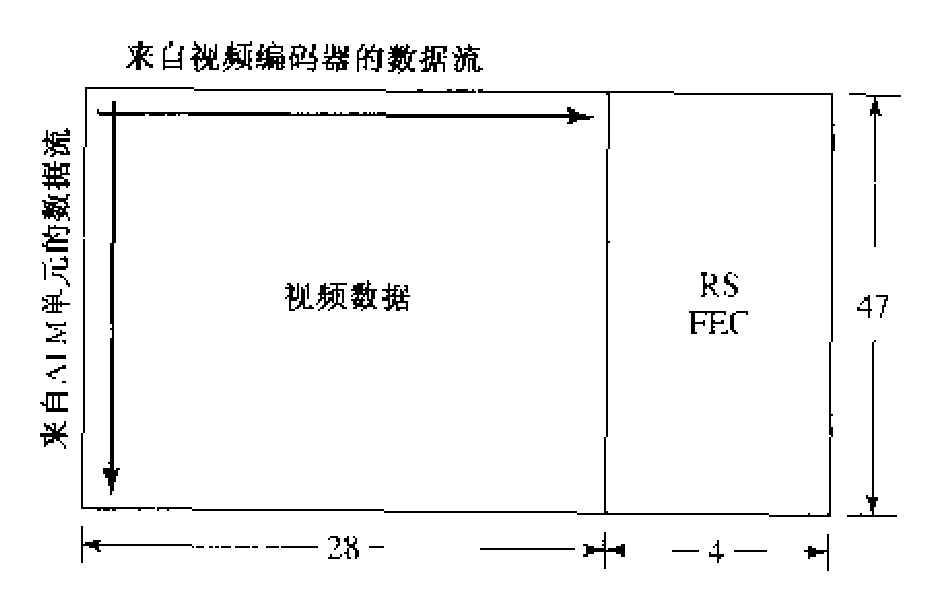


图 14.5 用于 ATM 单元丢失恢复的打包级 FEC 的图示

14.3.2 差错复原打包和复用

丢包对重建视频具有的影响取决于压缩的数据是如何打包的。以错误可隔离在一个小区域内的方式打包非常重要。这意味着不能盲目地把压缩的比特打成相同尺寸的包；而应该按照编码算法来构建包，以便包能包含一个或几个独立编码的数据块。例如，对于 MPEG-2 编码的视频，一个包可以包含一个或几个片(slice)，在每个包的开始处可能重复图像头信息。类似地，对于 H.263 编码的视频，一个包可包含一个或几个 GOB(块组)。在因特网中支持这种应用层已知的打包，称为应用层成帧(ALF)[13]。在支持包尺寸可变的网络中，希望使用最大可能的包，使由于包头和包处理开销所引起的有效码率下降最小。例如，在因特网上，最大传输

单元(MTU)大约是 1500 字节。这个尺寸对于包装来自片或 GOB 的数据足够大。对于低比特率的交互式应用系统,可接收的延迟会把最大包尺寸限制到一个比 MTU 小得多的值。在这种情况下,可用包头压缩来降低开销[13]。

如前所述,对于因特网上的视频传输,一般使用 RTP/UDP/IP 协议栈。RTP 通过定义 RTP 有效载荷格式[73,32,15,7]提供对各种视频编码格式的支持。有关 H.261,H.263 ,MPEG-1 和 MPEG-2 的有效载荷格式的详细描述参见参考文献[13]。

为了防止由于单个包的丢失造成邻近块的丢失,可采用交织打包;通过交织打包,来自相邻块或行的数据被放到分离的包中,如图 14.6 所示。这样,受损的块将被未受损的块所包围,易于在解码器中进行错误隐藏。注意,在传输层使用交织打包要求信源编码器只在被相继打包的块范围内进行块级的预测。这将稍微降低预测效率。同样,不得不重复某些包头以便使每个收到的包是可自解码的。例如,在图 14.6 所示的打包格式中,在两个包中应重复每帧的头信息(包含偶数行和奇数行数据)。

包1	包2	包3	包4	…
帧1的偶行数据	帧1的奇行数据	帧2的偶行数据	帧2的奇行数据	…

图 14.6　一个交织打包示例

影响视频通信差错特性的另一个因素是,视频如何与其他数据(音频、控制)复用以及如何保护包头。复用头信息中的一个错误可能导致视频数据被误传、引起在接收端大块数据的消失。为了防止这类事件发生,每个包一般都由一个长同步标志引导,同时用 FEC 对包头加重保护。一个成功的例子是复用标准 H.223[38],它是在 H.324 系统下为低比特率多媒体通信开发。H.223 为支持电路交换网络(例如有线和无线调制解调器)的多信源数据复用而设计。H.223 提供一个分层的多级复用结构,允许在复用器本身的抗误码健壮性与复用器所引入的开销之间进行可分级的折中。第 0 级用于有线环境,在包头上用短同步标志而不用 FEC。较高的级针对较易出错的环境(如移动无线调制解调器),使用较长的同步标志和较强的包头保护。

14.3.3　有延迟限制的重传

除了低层的差错控制机制(例如 FEC)、差错复原打包及复用外,上层传输协议可进行差错控制,一般采用 ARQ 形式,根据检测,对丢失的或过度延迟的包请求重传,如 TCP 中所做的那样。重传已非常成功地用于非实时数据传输,但由于会引入延迟,一般认为对于实时视频应用是不可接受的。事实上,情况并非总是这样。例如,对于洲际交互式业务,一次重传只增加大约 70 ms 的延迟,这是可以接受的[59]。对于单向实时视频应用,如因特网流视频和广播,延迟容限可进一步放宽到几秒钟,从而可进行若干次重传。重传也被认为不适用于多点视频会议,因为来自大量解码器的重传请求会使编码器不知所措。然而,当在多点会议中使用多点控制单元(MCU)时,编码器与 MCU 之间的通路以及 MCU 与解码器之间的通路只是简单的点对点。重传可分别用于这些通路。关于使用重传的另一个考虑是,它可能使问题变得更糟,因为它将增加更多的网络通信,从而进一步增加丢包率。然而,如果适当控制重传,就可以提高端到端的质量。例如,编码器

可降低其当前输出码率，以使编码器输出与重传数据之和保持在给定的总码率之下。

显然，对于实时应用系统，必须对重传加以限制，以使引入的延迟在可接受范围之内。不是像 TCP 那样，不确定地尝试重传以恢复丢失的包，尝试重传的次数可以由所期望的延迟而决定[53]。也可把分层编码(14.4.4 节)与赋予优先权的重传结合起来，把分层编码器中基本层的包放到重传队列的前面以增加基本层的重传尝试次数[64]。最后，在每一次单个重传尝试中可发送丢失包的多个副本，希望至少有一个及时通过[98]。可以根据包的重要性来决定重传尝试的次数和丢失包的重传副本数目。例如，基本层包可重传几次，每次有多个副本，而增强层包则可简单地放弃。另一个专门为流视频应用系统设计的限制重传方案将在第 15 章说明。

14.3.4 不平等差错保护

压缩视频比特流中的二进制比特并非同等重要。例如，在基于块的混合编码器中，图像头和其他辅助信息比块数据重要得多。这些重要比特应受到重点保护，以使它们以低得多的误码率传输。当在信源编码器中使用分层编码时(14.4.4 节)，传输控制器必须为不同的层分配适当的优先级，这是一种传输层控制形式。

不同网络可以用不同方法实现传输优先级。在 ATM 网络中，ATM 单元头中有一个比特表示其优先级。当出现通信拥塞时，网络节点可选择首先丢弃低优先级的单元。在无线网络中也可通过用不同的功率传输子流来实现传输优先权。在因特网中，最新的 RTP 规范定义了传输分层视频比特流的一般机制[63]。也有的网络通过区分业务来支持优先级传输[4]。无论在无线网还是在因特网中，都可以对不同的层使用不同的差错控制处理来实现优先级。例如，重传和较强的 FEC 可应用于基本层，而不重传和较弱的 FEC 可用于增强层。在极端情况下，可以对最重要的信息进行复制。例如，MPEG 视频建议对图像头信息和量化矩阵进行两次传输 [70]。提供不同级别保护的另一个方法是用不同的传输协议。例如，在参考文献[14]中所述的视频点播系统中，在会话前，用 TCP 传输极少量的高优先级数据(会话控制数据)；而在会话期间，对于低优先级数据(视频数据)采用 UDP。

14.4 差错复原编码

上一节描述了可以在传输层实行的各种差错控制机制。这些方法用于编码的视频流，试图检出、纠正错误以及(如果有必要且可行的话)重传受损的数据。即使应用这些方法也不可能完全纠正差错，因而解码器接收到的比特流仍可能含有差错(比特错误以及丢包)。在本节中，我们描述能产生对传输误码具有健壮性的比特流的信源编码方法，使得差错不会对解码器的工作产生不良影响以及导致重建视频质量不可接受的失真。

与编码效率最优的信源编码器相比较，这种编码器一般效率较低，因为在没有任何传输差错的情况下用较多的比特来获得相同的视频质量。这些额外的比特称为冗余比特①；引入这些比特是为了当比特流被传输错误损坏时提高视频质量。差错复原编码的设计目的是，在假定的信道环境下，对于给定的冗余量达到最佳的解码视频质量，或者在保持规定质量水平的同时使引入的冗余度最小。有许多把冗余度引入比特流的方法。某些技术有助于隔离差错(14.4.1 节)，而另外一些技术

① 这类比特也称为“开销”。

则使解码器能根据差错检测进行更好的错误隐藏(14.4.2节和14.4.1节)。还有另外一类技术,其目的是保证基本级别的质量,并且根据出现的传输错误提供适度的降质(14.4.4节和14.4.5节)。

14.4.1 错误隔离

压缩视频流对传输错误敏感的一个主要原因是视频编码器用 VLC 表示各种符号。在码字中的任何比特错误或比特丢失不仅使该码字不可解码,也使正确地接收的后续码字不可解码或解出错误符号。错误隔离技术顾名思义是试图把传输错误的影响隔离在一个有限的区域内。这经常可以通过在压缩比特流中放入"重同步标记"以及通过所谓"数据分割"技术来实现。二者都已吸收入 MPEG-4 和 H.263 标准中。

插入重同步标记　提高编码器差错复原能力的一个简单而有效的方法是周期性地插入重同步标记。这些标记被设计成容易与所有其他码字及其轻微扰动后得到的码字区别开来。通常在重同步信息后紧接着一些头信息(有关空间和时间位置或为解码后续比特所需要的其他信息)。这样,解码器可以通过检测重同步标记重新开始正确的解码。显然,插入重同步标记会降低编码效率。首先,这种标记越长和越频繁,所用的比特就越多。其次,使用重同步标记一般会中断图像内的预测机制,如 MV 或 DC 系数的预测,从而会增加更多的比特。但较长的和频繁插入的标记也会使解码器较快地重新获得同步,从而使传输差错只影响重建帧中较小的区域。因此,在实际视频编码系统中,使用相对长的同步码字。

数据分割①　没有任何其他差错复原工具的情况下,在出错的位置与后面的第一个同步标记之间的数据将不得不被丢弃。为了达到较好的错误隔离,两个同步点之间的数据可进一步划分成更小的逻辑单元,在它们中间使用二级标记。这样,在出错位置前的逻辑单元仍可被解码。二级标记可以比一级标记短,因为它们只需与紧接在它们前面的逻辑单元中的数据不同就可以。这种方法已用于 MPEG-4 和 H.263 标准的差错复原模式中,用这种方法把片或 GOB 中所有宏块的宏块头、运动矢量以及 DCT 系数放在分离的逻辑单元中。这样,如果在包含 DCT 系数的逻辑单元中发生错误,那么前面的逻辑单元中所包含的宏块头和运动信息仍是可解码的。

14.4.2 健壮的二进制编码

除了通过插入同步码字或把数据分割成独立段进行错误隔离外,也可直接修改二进制编码方法,以使产生的比特流对传输错误更具有健壮性。我们给出两种这类技术。

可逆的变长编码(RVLC)　在前面的讨论中,我们假设一旦发生错误,解码器就会丢弃所有的比特,直到识别出重同步码字为止。用 RVLC[72],解码器不仅能解码重同步码字之后的比特,也能从后向解码下一个重同步码字之前的比特,如图 14.7 所示。这样,用 RVLC,将丢弃较少的正确接收的比特,缩小传输错误所影响的区域。通过适当增加复杂度,提供前向和后向解码器输出之间的交叉校验能力,RVLC 也有助于解码器检测出用不可逆 VLC 所检测不到的错误,或者提供有关错误位置的更多信息,从而减少不必要丢弃的数据量。在 MPEG-4 和 H.263中都采用了 RVLC,与重同步标记插入和数据分割配合使用。

---

① 注意,这里所述的数据分割与 11.1.4 节中所描述的生成可分级比特流的数据分割不同。在那里,表示所有宏块的头、MV、也许还有前几个 DCT 系数的比特被放入基本层流中,而所有宏块的其余比特被放入一个或多个增强层流中。这里,两个重同步标记之间所包括的宏块中不同逻辑单元的数据被顺序地排列,生成单个比特流。

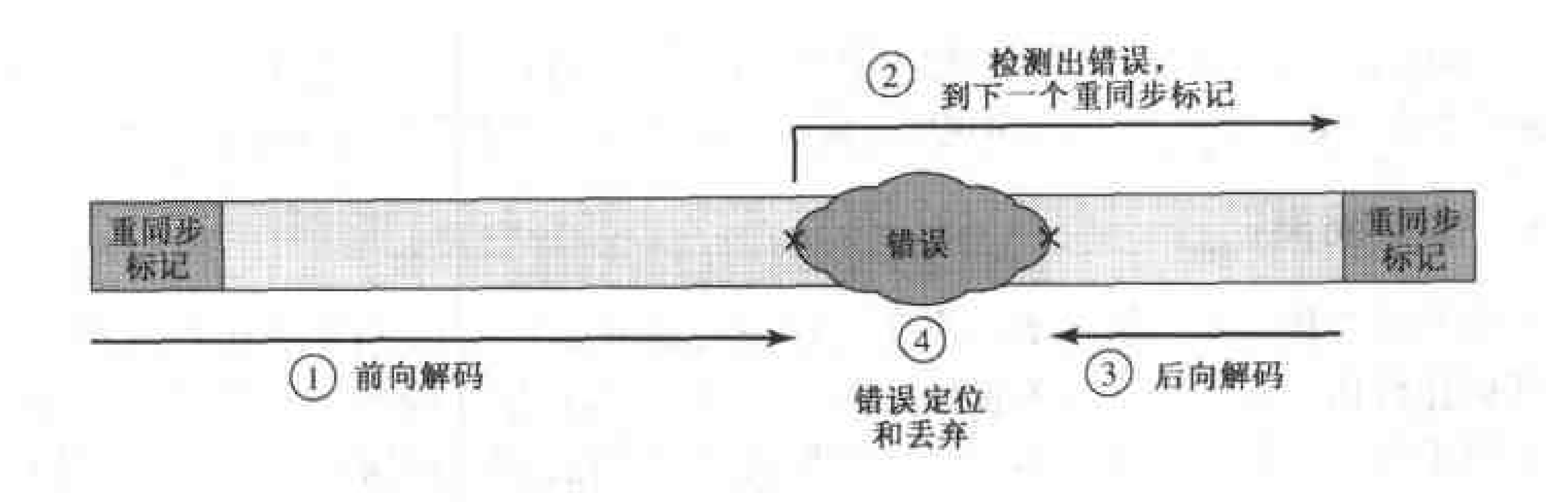


图 14.7 可以前向和后向解析 RVLC 码字，从而能够从受损的数据流中恢复更多的数据

虽然在设计变长码时有"可逆性"约束，但应用 RVLC 并不一定导致编码效率下降。事实上已经证明，压缩视频数据（例如已量化和游程编码的 DCT 系数，MV 差值）可由所谓的"广义高斯分布"(GGD)很好地建模，能够以接近理想的效率用高乐－瑞斯(Colomb－Rice,GR)和指数－高乐(exp-Golomb,EG)码进行熵编码。EG 码比 GR 码更有吸引力，因为即使在模型分布与实际数据不匹配时其性能也很稳定。因此，对于大多数视频序列，用固定的 EG 码表可能接近最佳的效率。GR 和 EG 码都有很好的结构。这使得不用查表法进行熵编码和解码以及设计有完全相同的编码效率的 GR 和 EG 码的可逆形式成为可能。关于如何设计与 GR 和 EG 码具有相同编码效率的 RVLC 表的更详尽描述在参考文献[86]中给出。除了提供前向和后向解码结果的交叉校验能力外，RVLC 比不可逆 VLC 能提供更好的检错能力（而且在某些情况下有更好的纠错能力）。关于可逆和不可逆 VLC 解码器工作的更详尽分析和比较在参考文献[85]中给出。

差错复原熵编码(EREC) EREC 方法不是用同步码字使解码器能重新开始解码，而是利用重排编码比特的方法使解码器在每个块的起始处重新获得同步[60]。具体地说，来自块组（或宏块）的可变长比特流被分布到相同尺寸的时间段上。起初，每个图像块的编码数据被全部或部分地放到为该块指定的时间段上。然后，用一个预定义的偏移序列寻找空的时间段，以放置比时间段大的那些块的任何剩余比特。这样做下去直到所有比特都包装到一个时间段中。由于每个时间段的尺寸是固定的，所以解码器在每个块开始时可重新得到同步。这也确保每个块的开始处比结束处更不受误差积累的影响。由于 EREC 不插入用于同步的任何比特，所以引入的冗余可忽略。EREC 在 MPEG-2 和 H.263 编码器中的应用已在参考文献[71,48]中考虑。

14.4.3 差错复原预测

正如我们已经看到的，VLC 的使用使压缩比特流对传输错误非常敏感。敏感的另一个主要原因是时间预测的使用。一旦发生误码，解码器重建的帧与编码器所呈现的不同，解码器所用的参考帧从该点往前就会与编码器所用的不同，从而所有后续的重建帧都将有差错。这种误差积累的影响已经示于图 14.2。对 DC 系数和 MV 使用空间预测也会引起误差积累，尽管误差积累限制在同一帧范围内。差错复原预测是指限制预测环从而把误差积累限制在一个短时间间隔内的技术。

插入帧内块或帧 阻止时域误差积累的一种方法是周期性地插入帧内编码图像或宏块。用 I 帧可能引起输出比特率的波动，因此由于延迟限制，对于交互式应用系统这一般是不可接受的。对于这类应用，使用足够多的帧内编码宏块是一种有效且可分级的差错复原工具。

当为了差错复原目的采用帧内编码宏块时，必须确定此类宏块的数目和它们的空间放置。

所需的帧内编码宏块的数目取决于信道质量和传输层所用的差错控制机制。许多实际系统提供关于网络质量的信息或获得此类信息的启发式方法。例证包括无线环境中的天线信号强度或关于因特网连接的 RTCP 接收器报告。参考文献[67]中的研究分析了解码器总失真对编码器的帧内编码宏块比率、信道编码速率(假设为里德-索罗蒙编码)以及信道误码参数(随机丢失率和突发误码长度)的依赖性。基于这种关系,对于给定的信道误码特性,可找到最佳的帧内编码宏块比率和(或)信道编码速率。

对于帧内编码宏块的空间放置,所建议的方法可分成启发式最优或率失真最优两种。启发式方法包括随机放置和高活动区放置。混合方案中还考虑了对于给定的宏块最后一次帧内更新的时间。这些方法简单且一般工作得很好。率失真优化方法可进一步改善性能,但以增加编码器复杂度为代价。

回想在没有传输误码的情况下,可以基于由不同模式获得的率失真折中来确定每个宏块的编码模式(9.3.3 节)。理想情况下,应该采用使每个比特对应失真度降低最大的模式。为差错复原的目的,可以采取同样的 RD 最优化方法,但编码器必须考虑的事实是,当计算与每个编码模式有关的失真时,当前的宏块和以前的宏块可能丢失。已在这种框架上提出了几种方法,但它们在计算解码器中所期望失真的方法以及信道误码模型化方法方面有所不同[17,94,89]。

例如,参考文献[17]中的算法取丢包率 p 的中期预测作为输入。每个宏块以帧内模式、帧间模式以及跳过模式进行编码;对于每一种模式,假设无损的宏块传输,测量产生的速率 $R(\mathrm{mode}_i)$ 和失真 $D_1(\mathrm{mode}_i)$。然后,对于同一组编码模式,在假设编码的宏块在传输期间受损的情况下,计算另一组失真 $D_2(\mathrm{mode}_i)$。在前面的失真测量中,要考虑在帧间编码宏块中误差积累的影响以及在解码器中对丢失宏块的错误隐藏。对于每一种编码模式,确定期望的失真 $D(\mathrm{mode}_i)=(1-p)\ D_1(\mathrm{mode}_i)+pD_2(\mathrm{mode}_i)$。确定得到最优的率失真折中的编码模式。这是通过对于不同模式、对于一个固定的 λ 确定拉格朗日函数 $D(\mathrm{mode}_i)+\lambda R(\mathrm{mode}_i)$,并且寻找具有最小拉格朗日极值的模式实现的。拉格朗日因子 λ 是基于目标比特率选择的。

最后,当可利用反馈信道时,可传送有关丢失或损坏的宏块数据信息,以启动发送端的帧内编码。在 14.6 节将讨论这些方案。

独立的分段预测 限制误差积累范围的另一种方法是把数据域分割成几个段,并只在同一个段内进行时间和空间预测。这样,一个段中的误码不会影响另一个段。例如,一个帧可以分为多个区域(例如一个区域可以是一个 GOB 或一个片),并且区域 1 只能由前一帧的区域 1 进行预测。在 H.263 中这称为独立分段解码(ISD)。

这一类中的另一种方法是在一段中包括偶索引的帧而在另一段中包括奇索引的帧。这样,偶帧只由偶帧进行预测。这种方法称为视频冗余度编码,已被纳入 H.263[87,88]中。它也可被认为是一种实现多描述编码的方法,将在 14.4.5 节中描述。

14.4.4 具有不平等差错保护的分层编码

分层编码(LC)是指把视频编码成一个基本层和一个或几个增强层。基本层提供低的但可接受的质量,而每一个附加的增强层递增地改善质量。如第 11 章所述,分层编码是可分级编码的一种特殊情况,它使有不同带宽容量或解码能力的接收机以不同质量访问同一个视频流。典型的实现包括 SNR 可分级、空间可分级以及时间可分级编码(见 11.1 节和 13.5.3 节)。为支持差错复原性能,分层编码必须与传输系统中的不平衡差错保护(14.3.4 节)结合使用,以

加强对基本层的保护。图 14.8 示出一般的两层编码和传输系统的方框图。

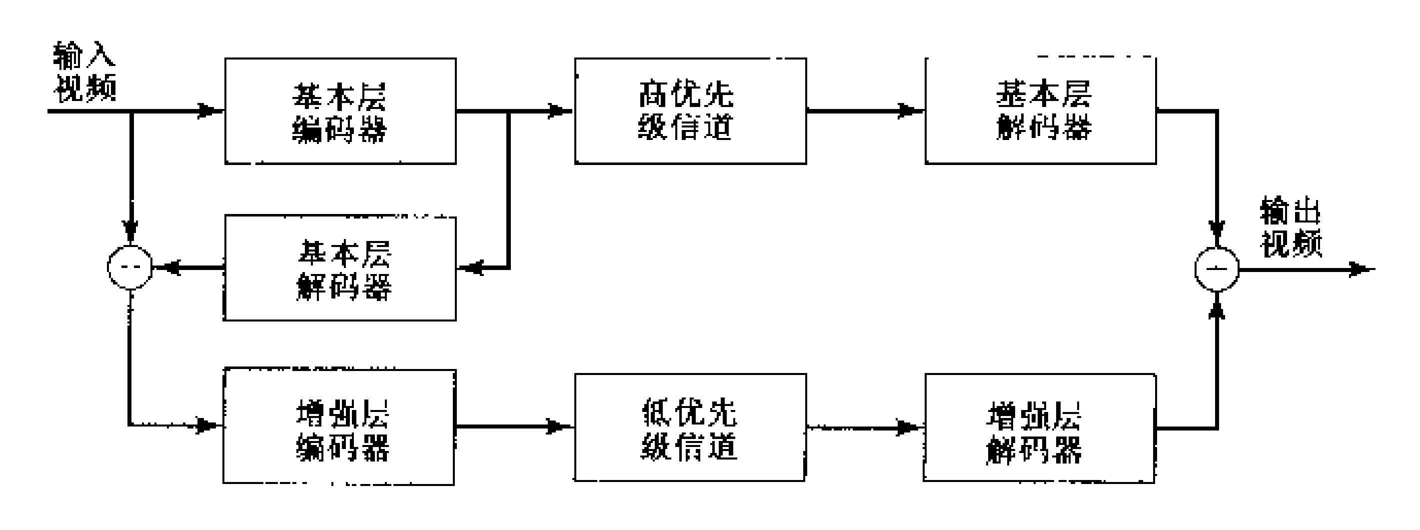


图 14.8　一个使用分层编码和优先级传输的系统的方框图

分层编码中的冗余度主要有两个来源。第一，为了避免误差积累，增强层可选择只用基本层的帧作为时域预测的参考帧。在这种情况下，将降低预测增益和随之而来的编码效率。第二，在每层中必须传输相似的边信息(例如头信息、编码模式以及 MV)。参考文献[1]中介绍了对 MPEG-2 中不同可分级模式的冗余度与差错复原性能之间所做的折中。对视频在无线网络上传输所用的分层编码和不平衡差错保护已经进行了广泛的研究；例如参考文献[95,46,31,24]。

14.4.5　多描述编码

当基本层在一个基本上无误码的信道中传输时，通过强 FEC 和重传前一小节所描述的分层编码可提供差错复原性能。然而，在某些应用中，保证任何数据的无损传输是不可行或是代价高昂的。在这种情况下，基本层中的损失可导致解码视觉质量灾难性的后果。抵抗传输错误的另一种方法是使用多描述编码(MDC)。用这种编码方案，生成同一个源信号的几个比特流(称为描述)并在分离的信道上传输。每个信道可能会暂时地中断或遭到长的突发误码。在信宿端，根据哪几个描述被正确地接收而启用不同的重建方案(或解码器)。MDC 编码器和解码器的设计使通过任何一个描述重建的信号的质量都是可接受的，而且用附加的描述可达到递增的改善。双描述编码器的概念如图 14.9 所示。在本例中，信宿有三个解码器，并且在任意给定的时间只有一个工作。

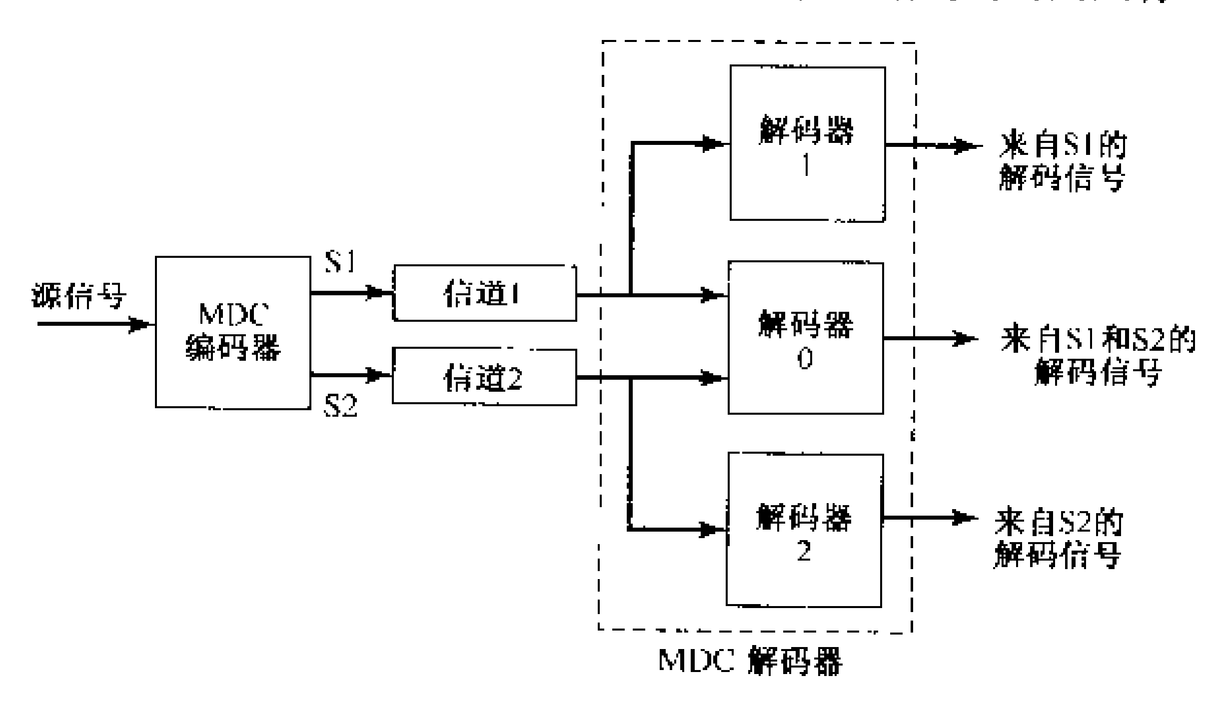


图 14.9　多描述编码和解码的图示

使用 MDC 时，传输不同描述的信道可以是信源与信宿间不同的物理途径，例如专用无线

网或因特网之类的包交换网络。即使当信源和信宿之间只存在一个物理路径时，该路径也可以用时间交织、频分复用等划分成若干虚拟的信道。例如，在因特网中，如果包尺寸相对较大，并且两个描述被放在交替的包中，那么两个描述的损失特性将接近于相互独立。

由于每种描述要提供可接受的质量，所有的描述都必须共享某些信源的基本信息，因此它们一定是相关的。这种相关性使解码器能由接收到的描述估计丢失的描述。另一方面，这种相关性也是 MDC 中冗余度的来源。MDC 比 LC 的优越之处在于它不需要网络的专门设备来提供可靠的子信道。例如在噪声很大的网络中，必须启用大量重传或者必须在 FEC 中添加大量的冗余以实现无差错传输。在这种情况下，用 MDC 可能更有效。

为了达到它们各自的目标，LC 用分层的、去相关的分解，而 MDC 用非分层的、相关的分解。已经提出了一些完成此类分解的方法，包括重叠量化[74,44,21]、相关预测器[34]、相关线性变换[80,28]、块相关重叠正交变换[11,12]、相关滤波器组[91,65,45]以及交织的空间－时间采样[79,87]。在 H.263 标准中，交织的时间采样方法称为视频冗余编码。下面，我们用相关线性变换更详细地描述这种方法。

多描述变换编码(MDTC)　在这种方法中，对原始信号样点进行线性变换以产生相关的系数组。理想情况下的变换应该是，变换系数可以被分成许多组，使不同组之间的系数是相关的。这样，如果在传输期间丢失了某些系数组，它们能够由接收到的组进行估计。为了使编码效率的损失最小，同一组内的系数应当是不相关的。为了简化有记忆信源信号的设计过程，可假设存在一个预白化变换，以便可对不相关的样点进行引入相关性的变换。

为了简化变换的设计，在参考文献[80]的方法中，把成对相关变换(PCT)应用于每一对不相关的系数。PCT 所产生的两个系数被分到两个流中，然后独立地进行编码。如果收到两个流，那么对每对变换系数应用逆 PCT，就可以精确恢复原始变量，只存在量化误差。如果只接收到一个流，基于两套系数之间的相关性，在丢失流中的系数可由接收到的流中的系数进行估计。图 14.10 示出了单个变量对的这种编码方案的方框图。这种方法所引入的开销可以由成对的系数个数、配对方法以及成对系数的变换参数进行控制。在固定冗余度下使单个描述的失真为最小的最优变换具有如下形式：

$$
T = \begin{bmatrix} \sqrt{\dfrac{\cot\theta}{2}} & \sqrt{\dfrac{\tan\theta}{2}} \\ -\sqrt{\dfrac{\cot\theta}{2}} & \sqrt{\dfrac{\tan\theta}{2}} \end{bmatrix}
$$

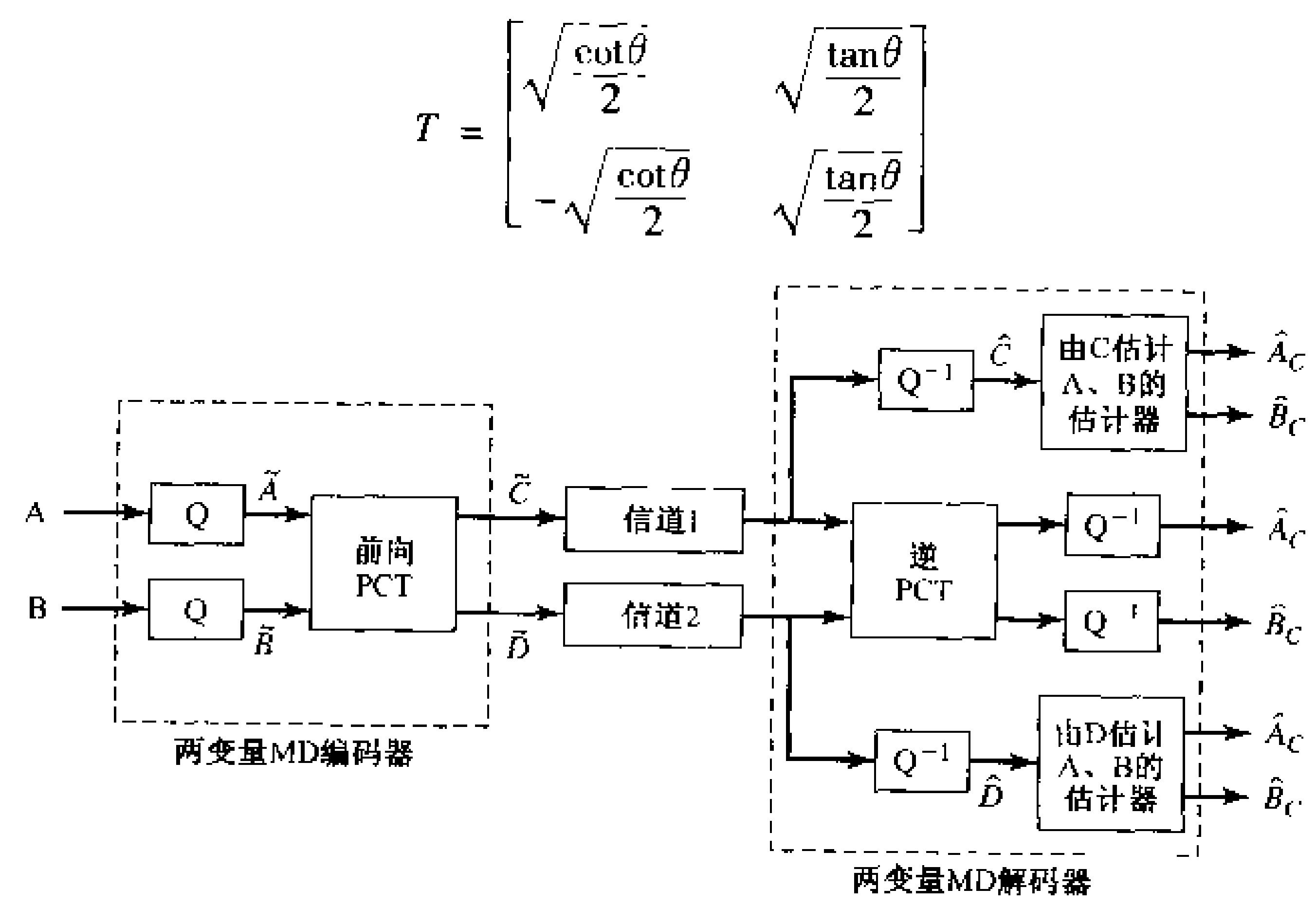


图 14.10　使用成对相关变换的多描述变换编码

参数 θ 控制由编码每对变量所引入的冗余度的大小。设要编码 $N \geq 2$ 个变量,存在一个最优配对策略,配合所选对之间的最佳冗余度,对于给定的总冗余度可使单个描述的总失真为最小。

MDTC 方法已纳入 JPEG 类的图像编码器中,其中,DCT(作为去相关变换)系数用 PCT 分成两个流。图 14.11 示出了在两个不同的冗余度下,由单个描述(即一半的比特被丢失)重建的图像;冗余度定义为在没有传输损失的情况下,为达到相同的失真,每个像素比参照的单描述编码器(在这种情况下为 JPEG 编码器)所需的额外比特数。可以看到,在相对低的冗余度下由单个描述可获得满意的图像。

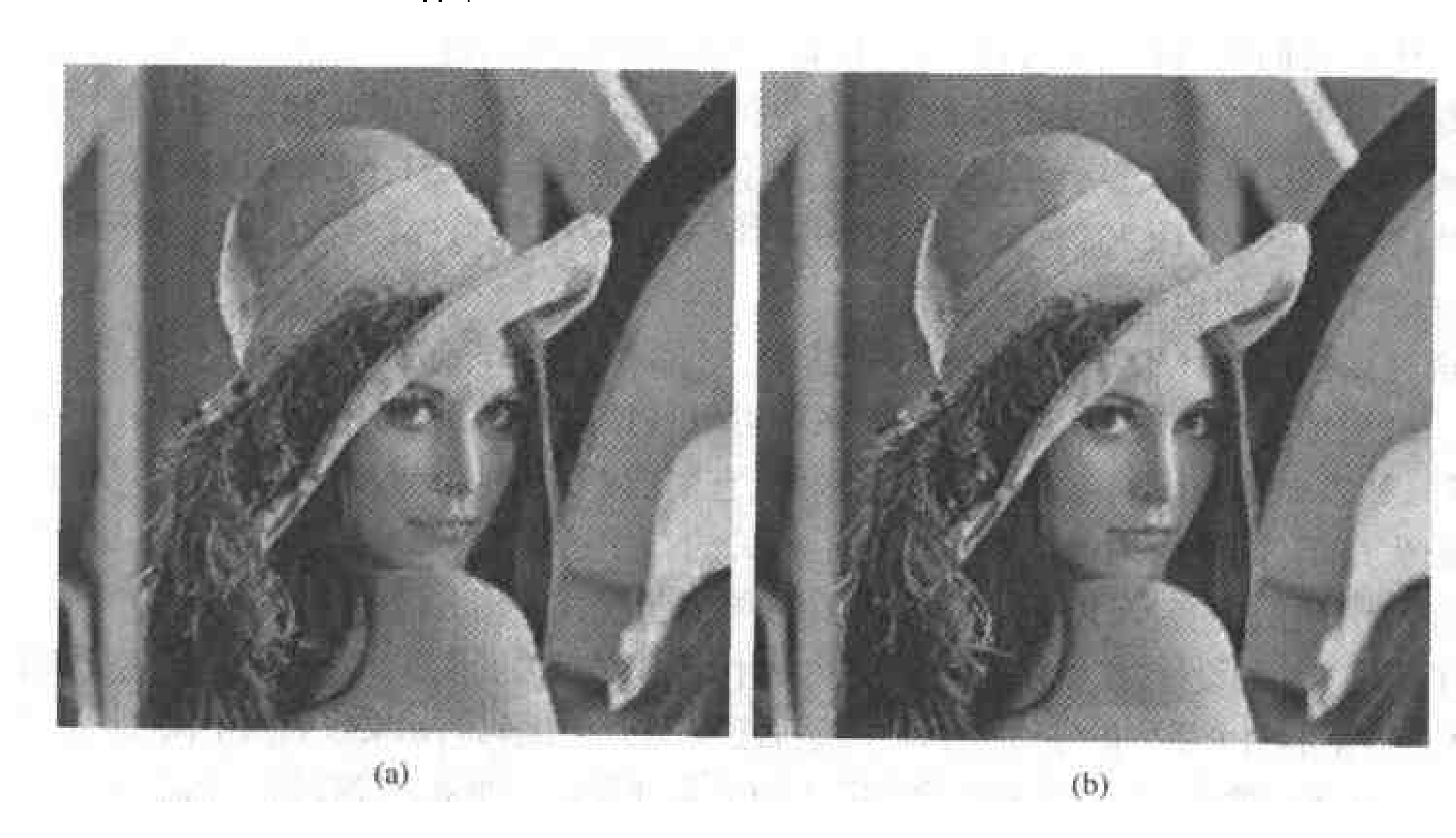


图 14.11 用多描述变换编码方法由单个描述重建的"莱娜"图像:(a)冗余度 = 0.088 bpp(15%),PSNR = 28.81 dB;(b)冗余度 = 0.133 bpp(22%),PSNR = 29.63 dB。当接收到两个描述时重建 PSNR = 35.78 dB。单描述编码器(JPEG)所需要的参考比特率为 0.60 bpp。这幅图像的原始未压缩版本已在图 9.10 中示出

MDTC 方法也已经加入基于块的混合视频编码器,其中用 MDTC 编码运动补偿预测误差以产生两个描述。在这种情况下,挑战性的问题是如何控制信源编码器中所用的参考帧与可能只接收到一个描述的解码器中所用的参考帧之间的失配。关于特殊的实现选项,见参考文献[61]。

当给定与 MDC 有关的相对大的开销时,这种方法只适合于丢失率相对高的信道。当信道丢失率小时,无误码情况下的重建性能起决定作用,因此更适于单描述编码。另一方面,当丢失率非常高时,有损情况下的重建质量更为关键,因此 MDC 方法更为适合。一个挑战性的任务是设计 MDC 编码器,使得它能根据信道误码特性自动地修改添加的冗余度。

14.4.6 信源和信道联合编码

在分层编码和 MDC 中,信源编码器和信道编码器之间在高层上交互作用。在分层编码中,信源编码器生成分层的流,假设信道编码器能够保证基本层的传输。另一方面,对于 MDC,信源编码器假设所有的编码比特将被同等地对待,并且所有比特都会受到损伤。信源-信道的交互作用也可以发生在较低的层上;例如,量化器和熵编码器是在信源编码器中设计

的,而 FEC 和调制方式是在信道编码器中设计的。虽然这种类型的方法传统上称为信源和信道联合编码,但广义上,分层编码和 MDC 也可认为属于此种类型。

用于压缩视频的一个流行的 FEC 方案是码率兼容删除卷积(RCPC)编码[29];用此种编码可容易且精确地控制信道冗余率。通过联合设计 RCPC 码和信源编码器中的比特率分配,可实现信源和信道编码速率之间所期望的折中。用 RCPC 码也可以相当容易地实现不平等差错保护[30]。

另一种信源和信道联合编码的方法是对于给定的信道误码特性联合地设计量化器和二进制编码器,以使传输误码的影响最小[49,19,56,57,75,20,78]。也可以仔细地设计已量化信源码字与它们的调制符号之间的映射。通过把调制星座空间中邻近的点分配给邻近的信源码字,可以减小信道误码的影响 [93,22]。

注意,此类技术主要针对比特错误,而在压缩数据被打包并且传输误码主要是由丢包引起的通信系统中并不是很有效。

14.5 解码器错误隐藏

如前几节中所提到的,由于各种信道或网络误码,在传输或存储期间压缩视频数据可能被损坏或丢失。在 14.3 节中,我们描述了可在传输层使用的使传输损失最小化的机制。除非可以无限次使用重传 (对于实时应用系统这是不可接受的),否则在接收的数据中仍然有差错或损失。在 14.4 节,我们给出了可以被编码器用来抑制这类差错影响的各种方法。在解码器的重建信号中传输错误仍然可导致令人讨厌的视觉失真。所引入的失真取决于许多因素(例如信源编码、传输协议、信息损失的数量和类型,等等),可从暂时的降质到图像或视频信号完全不可用。在本节中,我们介绍可以在解码器中使用的隐藏传输误码影响的方法,以使解码的信号视觉质量提高。

由于各种限制(例如编码延迟、实现的复杂度以及良好信源模型的可用性),尽管为达到最大可能的压缩增益已进行了大量的研究,压缩视频比特流仍具有一定程度的统计冗余,因此错误隐藏是可能的。另外,信源编码器可结合差错复原机制,有意把冗余引入编码比特中,以便于丢失数据的估计。此外,人类感知系统可以容忍一定程度的信号失真。所有这些因素可以用于解码器的错误隐藏。

错误隐藏属于图像恢复或修复的一般问题。然而,由于差错是在压缩比特级出现的,所以在像素域产生的错误图案非常特殊,通常需要采取特殊的措施来处理此类差错。如前所述,由于使用预测编码和 VLC,单独一个比特的差错就可使一个大的区域受到破坏。为了抑制差错的影响,可在解码器中采取各种措施,以一定程度的编码效率和复杂度为代价,使压缩流更具有差错复原能力。这里,我们假设同步码字被周期性地插入到一幅图像内(见14.4.1节),并且预测环路被周期地复位(见 14.4.3 节),使得一个比特的差错或丢包只引起一幅图像中有限区域内的损坏。

受损区域的范围取决于同步码字的频率、传输包尺寸以及比特率可由部分宏块到整个图像。对于如在 ATM 网络中的小信息包高比特率传输,一个丢失的包只会损坏部分宏块或几个邻近的宏块。在这种情况下,一个受损宏块一般被多个未受损宏块所包围。如果是这种情况,我们可用空间内插来恢复受损的宏块。另一方面,对于包相对大的低比特率应用系统(例如

IP 网的 128 kbps 比特率),一个丢失的包很可能损坏大部分帧或整个帧。在这种情况下,我们必须依靠前面和后面的视频帧来隐藏受损的帧。显然,即使在同一帧中相邻的宏块可用,也可结合相邻帧的信息。

对于基于块的混合编码模式,在一个受损的宏块中可能需要估计三种类型的信息:纹理信息,包括原始图像块或预测误差块的像素或 DCT 系数值;运动信息,由以 P 模式或 B 模式编码的宏块的运动矢量(MV)组成;以及宏块的编码模式。这些不同类型信息的恢复技术有所不同。

众所周知自然场景的图像主要具有低频分量,也就是说,除具有边缘的区域外,空间和时间上相邻像素的彩色值是平滑变化的。为恢复纹理信息开发的所有技术都利用了图像和视频信号的这种平滑性质,实际上它们都进行某种空间/时间内插。MV 域在较小程度上也利用了这种平滑性质,也可以用空间/时间内插来加以恢复。对于编码模式信息,所开发的方法更倾向于采用探试法。在以下小节中,我们将综述每类的一些代表性技术。这里所讨论的方法(以及其他方法)的细节参见参考文献[82,99,47]。我们将假设错误位置已经通过其他的方法被检测出来,例如在使用 RTP 的业务中基于序列号的丢包检测。关于各种差错检测方法的内容见参考文献[82]。

注意,在典型的(非差错复原)编码器实现中,这三种类型的数据比特是以每个宏块顺序存储的,使得它们在受损块中都会丢失。通过使用数据分割(14.4.1 节),能够使编码模式、MV、可能还有前几个 DCT 系数在受损块中仍是可利用的。如下所述,有关编码模式和 MV 的知识可以大大有助于纹理信息的恢复。

14.5.1 纹理信息的恢复

运动补偿时间内插 恢复受损宏块的最简单方法是复制前一个解码帧的对应宏块。然而,如果场景中有大的运动,结果并不令人满意。更有效的解决办法是使用由 MV 所指向的前一帧中的宏块。这种方法的恢复性能关键取决于 MV 的可用性。当 MV 也被丢失时,就必须首先估计 MV(14.5.2 节)。为了减少所估计的 MV 中误差的影响,可以把时域预测与空间内插结合起来。

空间内插 另一种简单的方法是由同一帧中相邻的正确接收的块中的像素来内插受损块中的像素。通常同一行的所有块(或宏块)被放在同一个包中,使得一个包的丢失一般会导致同一行中所有块的丢失。在这种情况下,可被受损块利用的相邻块只是它上面和下面的块。由于这些块的大多数像素离丢失的样点太远,所以通常只用相邻块的边界像素进行内插。一种较简单的方法并不内插各个像素,而是估计受损块的 DC 系数(即平均值),并且用等于所估计的 DC 值的常数来代替受损块。可以用周围块的 DC 值的平均来估计 DC 值。便于这种空间内插的一种方法是采用交织打包(14.3.2 节),使得一个包的丢失将只会破坏隔行的块或宏块。

最大平滑恢复(MSR) 空间内插方法所存在的问题是确定适当的内插滤波器。另一个缺点是它们忽略了所接收到的 DCT 系数。这些问题在参考文献[100,83]的 MSR 方法中得到解决,它要求受损块中恢复的像素与相邻的像素无论是在同一帧内还是在前/后帧都平滑地连接。如果接收到这个块的一些而不是所有的 DCT 系数,那么做出的估计应该使所恢复的块尽可能是平滑的,约束条件是所恢复块的 DCT 将产生与所收到的系数相同的值。这些目标可以形式化为一个最优化问题,而在不同丢失图案下的解决方案将对应于空间、时间和频率域的不同内插滤波器。

采用凸集投影(POCS)的空间内插 实现空间内插的另一个方法是用 POCS 方法[69,92]。基于 POCS 估计方法的一般思想是把每一个关于未知数的约束视为一个凸集。最优的

解是所有凸集的交,它可以通过递归地把前一个解投影到各个凸集上得到。当应用 POCS 恢复图像块时,空间平滑准则是在频域公式化的,要求恢复块的 DFT 只在几个低频系数中具有能量。如果认为受损块包含有某个特殊方向的边缘,那么可以要求 DFT 系数沿着与边缘方向正交的一个窄条分布,也就是说,沿着边缘方向低通而沿着正交方向全通。对每个 DCT 系数幅度范围的要求也可以转换为一个凸集。类似地,由任何接收到的 DCT 系数所施加的约束也形成一个分离的凸集。由于只能通过迭代程序获得解,所以这种方法不适于实时应用。

14.5.2　编码模式和运动矢量的恢复

正如所示,一些前述算法是以对编码模式和 P 或 B 模式宏块的运动矢量的知识为前提的。为便于实现解码器错误隐藏,编码器可进行数据分割,把模式和 MV 信息打包到分离的段中,并且用较强的差错保护传输它们(见 14.4.1 节)。例如,这是在 H.263 和 MPEG-4 中的一种差错复原模式。模式和 MV 信息仍是可能被损坏的。

估计受损宏块的编码模式的一种方法是收集相邻宏块编码模式的统计特性,并在给定周围宏块模式的条件下寻找最可能的模式[68]。当编码模式丢失时,一种简单和保守的方法是假设宏块以帧内模式编码,并只用空间内插恢复块。

要估计丢失的 MV,有几种简单的做法[50]:(1)假设丢失的 MV 为零,这对于具有相对小的运动的视频序列很有效;(2)采用前一帧中对应宏块的 MV;(3)采用空间相邻宏块的平均 MV;(4)采用空间相邻宏块的 MV 的中值;(5)重新估计 MV[33]。通常,当宏块受损时,其水平相邻的宏块也会受损,因此平均或均值是由上面和下面的 MV 得出。更简单的方法是采用受损宏块之上的宏块的 MV。研究表明,当采用一个以上 MV 的均值或中值时,图像质量不会达到明显的改善[81]。为了便于估计丢失的 MV,可用交织打包(14.3.2 节)方法。例如,如果宏块的奇数行和偶数行分别打包,那么包的丢失只损坏相隔的行。

14.5.3　基于语法的修复

前面的错误隐藏技术工作在信号域,假设已检测出传输流中的错误,已经丢弃“有问题”的数据单元,并且剩下的比特已经被 VLC 解码器解码成信号变量(例如边信息、MV、DCT 系数)。另一方面,基于语法的修复则是试图直接检测并修复比特流中的错误。这是可能的,因为对于任何有限长的数据包,只有相对少量的合法的码字组合。另外,压缩比特流决不会是一个独立恒等分布的二进制序列。在比特流中 1 和 0 的分布经常与位置有关[81]。基于这样的信息,解码器可以修复比特流,使得修复后的流符合编码方案所指定的语法。在 H.263 和 MPEG-4 中所用的数据分割和 VLC 有助于进行基于语法的修复,因为它们允许较多的“检查点”和语法限制,使合法的码字组合的总数小于相同长度的所有可能的二进制字符串的数目。一个称为“软解码”的这类修复方案在参考文献[84]中进行了描述。对于比特错误占主导地位的信道,基于语法的修复是一种有效的错误隐藏方法。对于可能丢失许多完整的包的丢包性网络,基于语法的修复只用于检测接收包的差错,而不能用于重建丢失的数据。

14.6　编码器-解码器交互的差错控制

到现在为止所提出的技术中,为抵抗传输误码,编码器和解码器都是独立工作的。设想可利用由解码器到编码器的后向信道,如果在差错控制过程中发送端和接收端协同工作,则可能达到较好

的性能。可以在信源编码或传输层中实现协同工作。在信源编码器中,可以基于来自解码器的反馈信息修改编码参数。在传输层,可利用反馈信息改变用于 FEC 或重传的总带宽的百分比。在14.3节中描述了传输层的适应性。在本节中,我们介绍几种基于来自解码器的反馈信息而修改信源编码策略的技术。这些技术开发的前提是差错持续时间不太长,存在一些差错也经常是可接受的情况。因此,即使不能做到纠正每个出现的错误,限制此类差错扩散的范围也是很重要的。

在描述这些技术前,应注意反馈信息是如何传送的。一般反馈消息不是视频语法的一部分,而是在协议栈的一个不同的层中传输,在这个层中交换控制信息。例如,在用于不可靠信道多媒体通信的标准 H.323 中[43],控制协议 H.245[41]允许报告受损宏块的时间和空间位置。一般,这些消息是无误码传送的,如有必要可以使用重传。

14.6.1 基于信道状况的编码参数自适应调整

在带宽和差错特性可变的信道中,重要的是使编码率与可用信道带宽相匹配,以及在编码比特流中嵌入适当的差错复原性能。当以高于信道可以及时传送的速率对信源编码时,根据网络层的判断将丢掉一些数据,通常会导致比以较低速率编码该信源所造成的信源编码失真更多的干扰效应。而且,当信道噪声很大时,最好是用较低的质量表示信源,留下更多的比特用于在编码流中以 FEC 或差错复原形式进行差错保护。

有两个问题必须解决:首先,传输控制器必须能基于反馈信息或其他传输层的相互作用,周期地估计并更新所建立连接的 QoS 参数(例如带宽、延迟、丢包率);其次,给定可用带宽和差错特性,编码器必须适当地确定编码参数(如在基于块的混合编码器中的帧内编码块比率、同步标记的频率、预测范围等),以便满足目标比特率和所期望的差错复原量。在 9.3.4 节中,我们讨论了码率控制问题,没有考虑差错复原因素。在 14.4.3 节中,我们描述了几种考虑了网络损失特性的编码模式选择方法(帧内与帧间)。Wu 等人的论文考虑了因特网视频传输的码率估计和码率自适应调整[90]。对于因特网上流视频应用系统,基于信道反馈调节目标编码率的有关内容将在第 15 章进一步讨论。

14.6.2 基于反馈信息的参考图像选择

利用可用的反馈信道的一个方法是使用参考图像选择(RPS)。如果编码器通过反馈信道了解到前一个编码帧的受损部分,它可以决定编码下一个 P 帧时不是采用最当前的参考图像,而是采用解码器中已知可利用的一个较早的参考图像。这要求编码器和解码器要存储多个前面已解码的帧。关于要用的参考图像的信息在比特流中传送。与将当前图像编码为 I 帧比较,如果所使用的较早参考图像不太远,则编码效率的下降并不很大。对使用 RPS 所引起的延迟和开销的研究可参见参考文献[26]。

注意,使用 RPS 不一定意味着编码器中额外的延迟。为编码当前的帧,编码器不必等待关于前一帧的反馈信息的到来,而是每当接收到反馈信息时,它可以选择受损帧前面的一个帧作为参考。例如,如果关于帧 n 的损坏信息直到编码帧 $n+d$ 时才到达编码器,那么在帧 $n+1$ 与帧 $n+d-1$ 之间的所有解码帧都会有差错,因为解码器使用与编码器不同的参考帧。在编码帧 $n+d$ 时,通过选择帧 $n-1$ 作为参考帧,误差积累将从帧 $n+d$ 开始中止。当然,生成和传送反馈信息所用的时间越长,编码效率的损失就越大。

14.6.3　基于反馈信息的错误跟踪

编码器可以不用较早的、未受损的帧作为参考帧，而是跟踪帧 n 中的受损区域如何影响帧$n+1$到帧 $n+d-1$ 中的解码像素。然后在编码第 $n+d$ 帧时按下述方法之一进行。编码器可以(1) 用帧内模式对由帧 $n+d-1$ 中受损像素预测的帧 $n+d$ 中的宏块进行编码；(2) 在编码帧 $n+d$ 时避免用帧 $n+d-1$ 中受到影响的区域进行预测；(3)对于帧 $n+1$ 到帧 $n+d-1$进行与解码器相同类型的错误隐藏，以便当编码帧 $n+d$ 时，编码器的参考图像与解码器的参考图像匹配。前两种方法只要求编码器跟踪受损像素或块的位置，而最后一种方法要求复制解码器从帧 $n+1$ 到帧 $n+d-1$ 的工作，比较复杂。用任意一种方法，解码器都将在帧$n+d$完全从误码中恢复。选择(1)所存在的问题是，如果帧 n 中有许多差错，将有太多的宏块需要以帧内模式编码。为了维持恒定的比特率，所有的宏块都需要较粗糙地量化。一个替代方法是只当帧 $n+d-1$ 的受损像素对帧 $n+d$ 中的宏块影响严重时，才启用帧内模式。这种方法不会在帧$n+d$立即阻止误差积累。因此，在后续的几帧中必须应用相同的补救措施[26]。这个方法如图 14.12 所示。有关错误跟踪、纠正以及快速算法的更多信息可见参考文献[77,18,26]。MPEG-4 ARTS类支持基于 NACK(否定应答)以帧内模式有选择地编码宏块(13.6.8 节)。

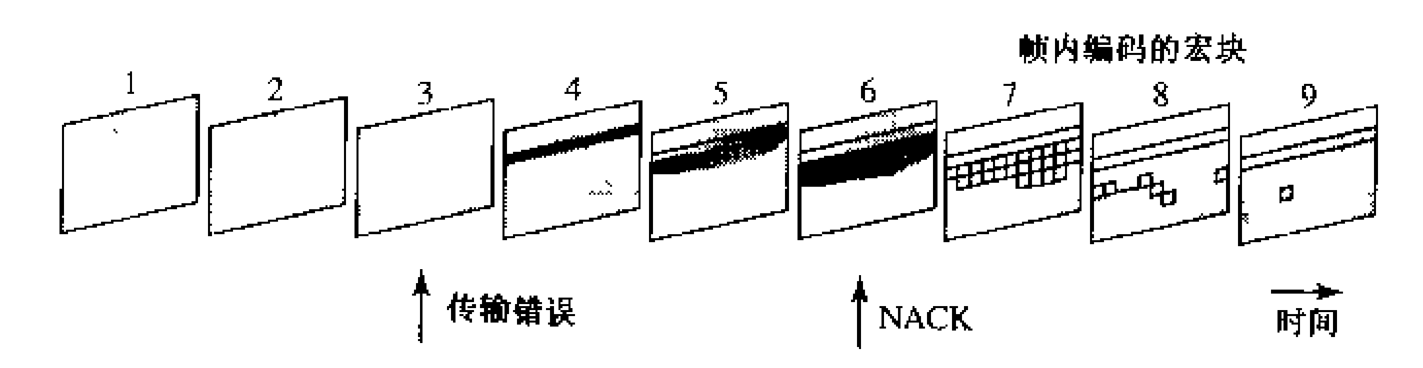


图 14.12　当采用错误跟踪并且受损块是根据收到的 NACK 信息进行帧内编码时的误差积累图示

14.6.4　无等待重传

为了利用重传数据，解码器的典型实现将不得不在处理相继收到的数据前等待所请求的重传数据到来。实际上，这是不必要的。用重传恢复丢失的信息而不引入延迟是可能的。在 Zhu[97,82]和 Chanbari[23]的方法中，当例如帧 n 的视频数据单元受损时，为恢复受损数据向编码器发送一个重传请求。不用等待重传数据到来，用所选择的错误隐藏方法隐藏受损的视频部分。然后，重新开始正常的解码，同时记录下受到影响的像素及相关的编码信息(编码模式和运动矢量)的跟踪情况(类似于 14.6.3 节所描述的错误跟踪方法)。根据例如在帧 $n+d$ 到来的重传数据，纠正受到影响的像素，使得它们好像没有发生传输差错一样被再生出来。纠正信号是由重传数据和所记录的跟踪情况得到的。

除了在信息丢失和重传数据到来之间的一段时间外，这种方法可实现无损恢复。在那段时间内，任何错误隐藏技术(14.5 节)都可用于受损区域。这种方案消除了与传统重传方案有关的延迟，而且不会损害视频质量。所付出的代价是相对高的实现复杂度。与 14.6.3 节中的修改编码操作以阻止误差积累的错误跟踪方法相比，这种方法利用重传的数据纠正并阻止解码器中的错误。

14.7 H.263和MPEG-4中的差错复原工具

在本节中,我们总结在H.263和MPEG-4标准中定义的差错复原工具。这些工具归入差错复原编码类,尽管标准只定义了比特流语法。值得注意的是,与任何差错复原编码方法一样,这些方法都会稍微降低编码效率。然而,它们为好的解码器提供了定位比特流中的错误、限制由这些错误影响的图像区域以及应用适当的错误隐藏方法(14.5节)的工具。如何用这些工具帮助解码器工作超出了标准的范围,这使得不同的解码器供应商之间能够相互竞争。

14.7.1 H.263中的差错复原工具

H.263标准的初始目标是作为H.324系统的视频编码标准,其目的是使视频电话能通过有线和无线调制解调器传输。如14.2节所描述的,适当地应用FEC和复用,可以认为有线和无线的调制解调信道是无差错的。因此,H.263开发的初始只是致力于提高编码效率。随着主要针对因特网上视频电话和视频会议的H.323标准的开发,差错复原成为主要关心的问题。开发了一组附件来扩展先前的H.263标准。以下是H.263标准版本3(即H.263++)中包含的差错复原工具,它于2000年11月被批准。

使用BCH码的前向纠错(附件H) 这个工具允许比特流中包含492个编码视频比特的块和19比特的BCH(511,492)FEC校验信息。加入一个用于重同步的附加比特组成512比特的块结构,附件H引入了大约4%比特率的开销。BCH(511,492)FEC码能纠正512比特块中的单个比特误码和可靠地检测2比特误码。这个工具是以ISDN信道上视频传输为目标的H.261标准的未用部分,目的是处理偶发性误码。对于易发误码的移动信道,附件H是无效的,因为既不能纠正也不能可靠地检测长于两个比特的突发误码。而且,492视频比特的固定块结构不能够使块边界与同步标记精确地对齐。由于这些原因,附件H对抵抗因特网或无线网的误码不是很有用[81]。

运用片结构模式的灵活的同步标记插入(附件K) 当使用这个工具时,用片结构取代GOB结构。片头作为同步标记,并且中断图像内MV和DC系数的预测。而GOB结构含有固定数目的宏块,每一个扫描次序的片大约有相同的比特数。如果在当前片中所包含的比特数超过了预定的阈值,那么在下个宏块的开始就生成一个新的片。如图14.13所示,视频中的活动区(其中每个宏块需要更多的比特)将比静止区具有更多的片,从而有更多的同步标记。这为解码器在活动区的错误隐藏提供了便利条件。

一个H.263中的片也可对应于帧中一个与宏块边界对齐的矩形区域。这使得能够进行如下所述的独立段解码。

参考图像选择(附件N和附件U) 如在14.6.2节中所介绍的,RPS允许编码器为运动补偿使用较早的参考图像,该图像是已知在解码器端被正确接收的。在H.263中,RPS可用于整幅图像、图像段(片或GOB)或单独的宏块。在附件N中定义了前两种机制,并且只是作为一种差错复原工具引入的,而后者定义在附件U中,是为了同时达到差错复原和编码效率而设计的①。在H.263中,无论有无反馈信息都可以使用RPS。当没有反馈信息时,可用14.4.3节中所描述的视频冗余度编码

① 通过允许宏块在一组过去的帧中间选择最佳匹配的宏块,以运动估计复杂度为代价可使编码效率提高。

方法，利用预先处理的交织的 RPS 方案。然而，此技术比基于反馈的机制效率低得多[87,88]。

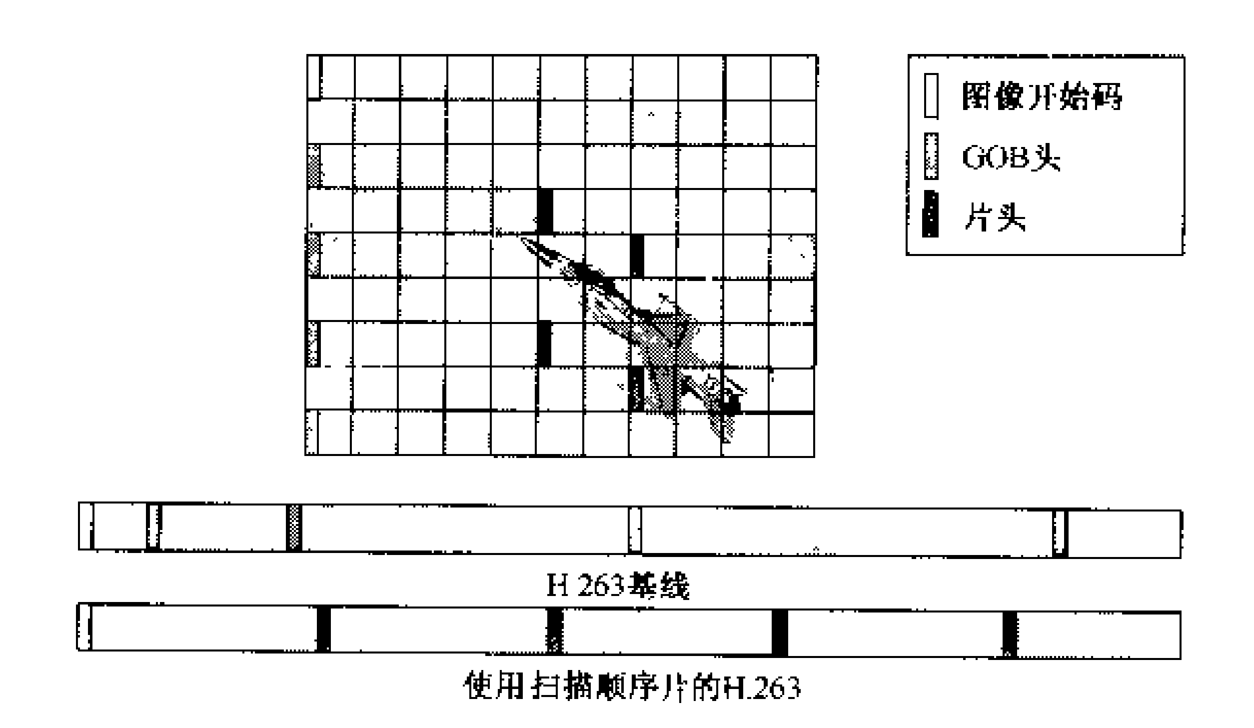


图 14.13　在 H.263 中扫描顺序片与 GOB 的对比。对于 GOB 结构，包含着运动飞机的 GOB(第二和第四个GOB)更有可能出错，因为它们含有较多的比特。受损的GOB将导致视觉上不可接受的失真。对于片结构，相同的区域被更多的片覆盖。因为在这个区域出现大量的同步标记，并且丢失的片影响较小的区域，使差错复原能力得到改善

可分级性(附件 O)　附件 O 定义了类似于在 MPEG-2 中所用的时间、空间以及 SNR 可分级性。如 14.4.4 节所述，如果在信源与信宿之间利用多条具有不同 QoS 特性的传输路径，则这些工具可用来提高差错复原能力。目前的网络通常不支持不同的 QoS，因此，可分级性很少为差错复原的目的在实际中使用。

独立段解码(ISD)(附件 R)　ISD 强迫编码器和解码器像对待图像边界那样对待段(片或 GOB)边界，借以阻止受损数据从一个段扩散到另一个段。这种方法是 14.4.3 节所讨论的差错复原预测的一种特殊情况。已经证明与矩形片相联系的 ISD 可提高差错复原能力[88]。ISD 的开销大约与图像尺寸成反比，对于小于 CIF 的图像尺寸，开销过高而不切实际[81]。

数据分割和 VLC(附件 V)　如前所述，对于数据分割，宏块头、MV 以及 DCT 信息不再逐个宏块地进行交织，而是用专门设计的标记把它们分组为不同部分。头和运动信息用 RVLC 进行编码，而 DCT 系数用 H.263 基本系统中的不可逆 VLC 表进行编码。

头重复(附件 W)　这允许在当前帧的头中重复前一帧的头，在允许延迟一帧的情况下，它使得能够对丢失了帧头的帧进行解码。注意，H.263 不包含允许在片级包含冗余的图像头信息的语法元素，可以在 MPEG-4 的头扩展码中得到该元素。

关于如何能够把上述工具纳入视频编解码器，并且用于因特网和无线网上视频传输的讨论，见参考文献[81]。

14.7.2　MPEG－4 中的差错复原工具

MPEG-4 标准的主要目标之一是使通用的访问成为可能。因此，差错复原从一开始就是一个所要考虑的重要问题。因为开发时间交叠，所以定义在 MPEG-4 中的差错复原工具相当类似于 H.263 中的那些工具。它包括前几节所描述的错误隔离、同步、数据恢复以及差错复原熵编码等许多工具。

重同步工具　MPGE-4 定义几种在传输差错发生后进行快速重同步的方法。视频打包方法非常类似于 MPEG-2 的自适应片和 H.263 的片结构模式。视频包的作用与 H.263 的扫描顺序片的作

用相同。视频包的长度不再基于宏块的数目(如MPEG-4的非差错复原模式或H.263基本系统的情况),而是基于该包所含的比特数(见图14.13)。它的目的是为比特流提供周期性的重同步。

重同步标记放在每个视频包头的开始位置。像其他起始码一样,这个标记有别于所有其他可能的VLC码字。头包含为重新开始解码过程所需要的信息,包括这个包的第一个宏块的地址和这个宏块的量化参数(QP)。它们的后面跟着单个比特的头扩展码(HEC)。如果HEC置为1,那么已经在视频对象平面(VOP)头中说明的信息被复制到这个包头中,如定时信息、时间参考、VOP预测类型以及一些其他信息。头扩展特性使解码器即使在包含VOP头的包被丢失的情况下,也能正确地使用包含在当前包中的数据。它也能进行交叉校验,因为在同一个VOP中的所有包应该共享相同的QP、时间戳,等等。

除了使用可变长视频包外,MPEG-4也采纳了一种称为固定间隔同步的方法。这种方法要求视频包的起始只在比特流中允许的、固定间隔的位置出现。这有助于避免由传输差错引起的与起始码竞争有关的问题。

数据分割 为了实现视频包中较好的错误隔离和固定间隔同步方法,MPEG-4也用数据分割作为差错复原工具。视频包内的数据分割重新组织数据,使得运动矢量和相关的语法成分被首先传输,其后跟着如CBP和DCT系数之类的语法成分。图14.14示出了具有数据分割的视频包的语法结构。注意,该结构中的DCT纹理部分可用RVLC编码。

一个视频包								
重同步标记	宏块索引	量化信息	HEC和头重复(如果存在)	运动矢量	运动标记	纹理(DCT)	重同步标记	…

图14.14 MPEG-4中具有数据分割的差错复原模式中的视频包的语法结构

NEWPRED模式 这个工具非常类似于H.263 RPS模式(附件N)和片结构模式(附件K)。当NEWPRED(ND)模式开通时,编码器用于时间预测的参考将按照来自解码器的反馈信息自适应地更新。这些消息指出哪些NEWPRED段(它可以是整个帧,或者——用MPEG-4术语——一个VOP,或者是一个包的内容,一般是一个片)没有被成功地解码。基于反馈信息,编码器将使用最近的ND段或者空间对应但时间较早的ND段来进行预测。

RVLC及数据分割配合使用的数据恢复 MPEG-4差错复原模式使用RVLC来实现更好的DCT数据恢复。在VOL上标识RVLC的使用。当RVLC用于DCT数据时,比特流首先被前向解码。如果没有检测到差错,那么比特流就被认为是有效的。然而,如果检测到差错,则用双向解码,并且不应该使用该包前几个宏块之间在前后两个方向都检测到差错的部分。

14.8 小结

在本章中,我们介绍了能够用于抵抗实时视频通信中传输错误的各种差错控制机制。重点介绍了为基于块的混合编码器开发的技术,特别是那些被H.263和MPEG-4标准所采用的技术。这里对不同种类的差错控制技术的主要特性归纳如下。

传输层差错控制(14.3节)

- 传输层差错控制机制是最重要的,并且保证基本的质量水平。可以通过FEC、交织打包

以及(若可行的话)有限制的重传实现差错控制。

- 可对具有不同重要性的编码比特部分应用不同级别的保护。这种不平等差错保护是用有限量的冗余度达到可接受质量的一种有效方法。

差错复原编码(14.4 节)

- 这类技术通过在信源编码器的编码比特流中加入一定量的冗余度实现差错复原。一些技术的目的在于保证基本的质量,并根据出现的传输差错提供适度的降质(例如分层编码和 MDC);一些技术有助于阻止误差积累(例如差错复原预测);而另一些技术有助于检测或恢复比特错误(例如错误隔离和健壮的熵编码)。一些技术要求信源编码器和传输层之间密切的相互作用(例如具有不平等差错保护的分层编码、具有交织打包的差错复原预测),而其他的技术假定不同的子流在传输层中是平等对待的(例如 MDC)。
- 如在 14.4.6 节中所述,错误隔离、RVLC 以及信源和信道联合编码只对抵抗比特错误有用,而无助于恢复丢包。

解码器错误隐藏(14.5 节)

- 所有的纹理错误隐藏技术利用图像和视频信号时间和空间的平滑性质恢复丢失的信息。MSR 技术通过使重建信号的粗糙度达到最小来实施平滑性约束。运动补偿的时间内插和空间内插方法可认为是 MSR 方法的特殊情况,其中只使时间或空间的残差度量达到最小。另一方面,POCS 方法把重建的图像块迭代地投影到由接收的系数和由估计的块的边缘方向决定的平滑约束所确定的凸集上。尽管一般会给出比 MSR 更精确的结果,但是 POCS 方法的计算量较大,因为它需要许多次迭代。
- 简单且有效的运动补偿时间内插已经在商业系统中被采用。只当场景有非常复杂的运动或正在经历场景改变时,利用空间相关的更复杂的方法(空间内插、MSR 和 POCS)才能进一步改善性能。
- 编码模式和运动信息的恢复比纹理信息更困难,因为在编码模式以及邻近宏块的运动矢量之间相关性很小。如果可能的话,通过使用数据分割和不平等差错保护,可以使这类信息比纹理信息更好地被保护起来。

编码器–解码器的交互式差错控制(14.6 节)

- 只有当存在从解码器到编码器的后向信道时,才可应用这个技术,使得解码器能够通知编码器丢失了哪部分编码信息。
- 所提出的三个技术(参考图像选择、错误跟踪和无等待重传)都是为了在接收到反馈信息后阻止误差积累。参考图像选择和错误跟踪避免编码器使用受损的区域进行未来的预测;而无等待重传基于重传的信息纠正差错。
- 如果只能利用一般信息,例如丢包率,而并不知道哪部分信息丢失时,可尝试调节编码参数,以便减小数据丢失并抑制任何丢失数据的影响。

技术的选择

- 传输层差错控制机制是最重要的,并且保证基本的 QoS 水平。为了进一步改善非常有损环境下(如因特网和无线网络)的质量,差错复原编码和解码器错误隐藏是必要的。取决于信道差错特性以及系统配置和要求,一些技术比另一些技术更有效。
- 突发性传输差错对算法选择具有相当大的影响。对于具有很长的突发误码的信道,差错复原编码技术可能不适用。这是因为这种编码技术所引入的冗余当信道无误码时是

浪费的，而当出现突发误码时，这种冗余又不是很有用。当适当限制时，重传可能更适用，因为它只在需要时引入开销。

- 从解码器到编码器后向信道的存在也影响一些方案的配置。在如广播之类的应用系统中，其中没有后向信道，不能应用任何交互式差错控制技术。类似地，在视频流应用系统中，视频一般预先压缩并存储在服务器中，因此不可能基于网络状况修改编码器的工作。
- 在任何情况下都可应用错误隐藏技术。然而，这种技术的有效性受到可用信息的限制。另外，某些技术可能对于有成本效益的实现太复杂，或者对实时应用系统引入不可接受的处理延迟。
- 除了延迟和复杂性问题外，比较不同方案的一个重要准则是为达到相同程度的差错保护所需要的信源编码器和传输层总冗余量。显然，需要最小冗余量的方法在带宽利用方面是最有效的。

14.9 习题

14.1 传输音频和视频与传输数据文件相比主要有什么不同的要求？

14.2 对于交互式视频会议应用系统的典型延迟要求是多少？对视频流呢？

14.3 影响在视频信源与接收端显示的视频之间端到端延迟的因素是什么？

14.4 我们把视频通信中的差错控制方法分为四种。它们是什么？请描述每一种的优点和局限性。

14.5 线路交换网络与基于包交换的网络之间的主要区别是什么？在每种情况下你选择什么作为你的差错控制机制？

14.6 分层编码(LC)与多描述编码(MDC)之间的主要区别是什么？在网络需求方面它们有何不同？叙述宁愿选择 LC 而不是 MDC 的网络状况，以及反过来宁愿选择 MDC 而不是 LC 的网络状况。

14.7 如果网络由可靠(几乎无误码)改变到不可靠(具有高的丢包率)，在实现运动补偿预测中你会采取怎样不同的做法？

14.8 从理论上说，丢失的信息是不可恢复的。是什么使错误隐藏技术对视频是可行的？

14.9 在恢复受损纹理数据的错误隐藏技术中的主要假设是什么？

14.10 描述错误隐藏的 MSR 和 POCS 方法的原理。每种方法的优点和缺点是什么？

14.11 考虑因特网视频会议应用系统。假设压缩的视频流被打包，并且包或者丢失（由于过长延迟或错传，或检测到包中有误码）或者被完整无缺地接收到。假设有效丢包率相当高，大约 10%，且无反馈通道可利用。信源与信宿之间的端到端平均延迟大约是 50 ms。进一步假设信源与信宿之间允许的最大端到端延迟是 150 ms。你会提出什么作为控制并隐藏丢包影响的机制？本章中所述的哪些方法是不适用的或是有效的？

14.12 重新考虑习题 14.11 中的情况，但假设网络有较低的丢包率，例如 1%。你的解决办法会不同吗？

14.13 重新考虑习题 14.11 中的情况，但这次假设信宿与信源之间的反馈信道是可利用的，并且一般花费 50 ms 传送反馈信息。在这种情况下，你会提出什么解决办法？在反馈信道上你会传送什么？

14.14 现在考虑在具有高比特错误率、突发性错误和窄带宽(例如 64 kbps)的无线网络上的视频传输。即使网络或解码器已在基于 FEC 编码的包内检测到比特错误,你也有进一步检验该包的选择权。在这种情况下,你会提出什么解决办法?与因特网的情况相比较,你会用较短的包还是较长的包?

14.10 文献目录

[1] Aravind, R., M. R. Civanlar, and A. R. Reibman. Packet loss resilience of MPEG-2 scalable video coding algorithms. *IEEE Trans. Circuits Syst. for Video Technology* (Oct. 1996), 6(5):426–35.

[2] Ayanoglu, E., R. Pancha, A. R. Reibman, and S. Talwar. Forward error control for MPEG-2 video transport in a wireless ATM LAN. *ACM/Baltzer Mobile Networks and Applications* (Dec. 1996), 1(3):245–58.

[3] Berger, A. W. ATM networks. In M. T. Sun and A. R. Reibman, eds., *Compressed Video over Networks*. New York: Marcel Dekker, 2001, 139–76.

[4] Blake, S., et al. *An architecture for differentiated services*, IETF RFC2475, Dec. 1998. (Available at http://www.ietf.org/rfc/rfc2475.txt).

[5] Bojkovic, Z. S., and K. R. Rao. *Packet Video Communications over ATM Networks*. Englewood Cliffs, NJ: Prentice Hall, 2000.

[6] Bolot, J. C., H. Crepin, and A. Vega-Garcia. Analysis of audio packet loss in the Internet. In *The 5th International Workshop on Network and Operating System Support for Digital Audio and Video* (April 1995), 163–74.

[7] Bormann, C., et al. *RTP payload format for the 1998 version of ITU-T recommendation H.263 video (H.263+)*, IETF RFC2429, Oct. 1998.

[8] Budagavi, M., W. R. Heinzelman, J. Webb, and R. Talluri. Wireless MPEG-4 video communication on DSP chips. *IEEE Signal Processing Magazine* (Jan. 2000), 17:36–53.

[9] Chang, L.-F. Wireless systems and networking. In M. T. Sun and A. R. Reibman, eds., *Compressed Video over Networks*. New York: Marcel Dekker, 2001, 177–214.

[10] Chapplapali, K., et al. The grand alliance system for US HDTV. *Proceedings of the IEEE* (Feb. 1995), 83(2):158–74.

[11] Chung, D., and Y. Wang. Multiple description image coding using signal decomposition and reconstruction based on lapped orthogonal transforms. *IEEE Trans. Circuits Syst. for Video Technology* (Sept. 1999), 9(6):895–908.

[12] Chung, D., and Y. Wang. Lapped orthogonal transform designed for error resilient image coding. In *IEEE Int. Conf. Image Proc. (ICIP2000)*, 3:368–71. Vancouver, Canada, Sept. 2000.

[13] Civanlar, M. R. Internet video. In M. T. Sun and A. R. Reibman, eds., *Compressed Video over Networks*. New York: Marcel Dekker, 2001, 433–64.

[14] Civanlar, M. R., and G. L. Cash. A practical system for MPEG-2 based video-on-demand over ATM packet networks and the WWW. *Signal Processing: Image Communications* (1996), 8(3):221–27.

[15] Civanlar, M. R., G. L. Cash, and B. G. Haskell. *RTP payload format for bundled MPEG*, IETF RFC2343, May 1998.

[16] Comer, D. E. *Internetworking with TCP/IP*. Vol. 1: Principles, Protocols, and

Architecture. 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 1995.

[17] Côté, G., S. Shirani, and F. Kossentini. Optimal mode selection and synchronization for robust video communications over error prone networks. *IEEE Journal on Selected Areas in Communications* (June 2000), 18(6):952–65.

[18] Steinbach, E., N. Färber, and B. Girod. Standard compatible extension of H.263 for robust video transmission in mobile environments. *IEEE Trans. Circuits Syst. for Video Technology* (Dec. 1997), 7:872–81.

[19] Farvardin, N., and V. Vaishampayan. Optimal quantizer design for noisy channels: An approach to combined source-channel coding. *IEEE Trans. Inform. Theory* (Nov. 1987), 38:827–38.

[20] Fischer, T. R. Joint trellis coded quantization/modulation. *IEEE Trans. Commun.* (Feb. 1991), 39:172–76.

[21] Fleming, M., and M. Effros. Generalized multiple description vector quantization. In *Data Compression Conference (DCC'99)*, Salt Lake, UT, March 1999.

[22] Fuldseth, A., and T. A. Ramstad. Combined video coding and multilevel modulation. In *IEEE Int. Conf. Image Proc. (ICIP'96)*, 77–80, Lausanne, Switzerland, Sept. 1996.

[23] Ghanbari, M. Postprocessing of late cells for packet video. *IEEE Trans. Circuits Syst. for Video Technology* (Dec. 1996), 6(6):669–78.

[24] Gharavi, H., and S. M. Alamouti. Multipriority video transmission for third-generation wireless communication systems. *Proceedings of the IEEE* (Oct. 1999), 87:1751–63.

[25] Girod, B., and N. Färber. Wireless video. In M. T. Sun and A. R. Reibman, eds., *Compressed Video over Networks*. New York: Marcel Dekker, 2001, 465–512.

[26] Girod, B., and N. Färber. Feedback-based error control for mobile video transmission. *Proceedings of the IEEE, Special Issue on Video for Mobile Multimedia* (Oct. 1999), 87:1707–23.

[27] Goodman, D. J. *Wireless Personal Communications Systems*. Reading, MA: Addison-Wesley, 1997.

[28] Goyal, V. K., J. Kovacevic, R. Arean, and M. Vetterli. Multiple description transform coding of images. In *IEEE Int. Conf. Image Proc. (ICIP'98)*, 1:674–78, Chicago, Oct. 1998.

[29] Hagenauer, J. Rate compatible punctured convolutional codes (RCPC codes) and their applications. *IEEE Trans. Commun.* (April 1988), 36:389–99.

[30] Hagenauer, J., and T. Stockhammer. Channel coding and transmission aspects for wireless multimedia. *Proceedings of the IEEE* (April 1988), 87(10):1764–77.

[31] Hanzo, L. Bandwidth efficient wireless multimedia communications. *Proceedings of the IEEE* (July 1998), 86:1342–82.

[32] Hoffman, D., G. Fernando, V. Goyal, and M. R. Civanlar. *RTP payload format for MPEG1/MPEG2 video*, IETF RFC2250, Jan. 1998.

[33] Hong, M. C., L. Kondi, H. Scwab, and A. K. Katsaggelos. Video error concealment techniques. *Signal Processing: Image Communications, Special Issue on Error Resilient Video* (1999), 14:437–92.

[34] Ingle, A., and V. A. Vaishampayan. DPCM system design for diversity systems with applications to packetized speech. *IEEE Trans. Speech and Audio Processing* (Jan. 1995), 3:48–57.

[35] ITU-T. Recommendation H.261: Video codec for audiovisual services at $p\times64$ kbit/s, 1993.

[36] ITU-T. Recommendation H.324: Terminal for low bitrate multimedia communication, 1995.

[37] ITU-T. Recommendation G.114: International Telephone connections and circuits—General recommendations on the transmission quality for an entire international telephone connection—One way transmission time, 2000.

[38] ITU-T. Recommendation H.223: Multiplexing protocol for low bit rate multimedia communication, 1996.

[39] ITU-T. Recommendation H.221: Frame structure for a 64 to 1920 kbit/s channel in audiovisual teleservices, 1997.

[40] ITU-T. Recommendation H.320: Narrow-band visual telephone systems and terminal equipment, 1997.

[41] ITU-T. Recommendation H.245: Control protocol for multimedia communication, 1998.

[42] ITU-T. Recommendation H.263: Video coding for low bit rate communication, 1998.

[43] ITU-T. Recommendation H.323: Packet-based multimedia communications systems, 1998.

[44] Jafarkhani, H., and V. Tarokh. Multiple description trellis coded quantization. *IEEE Trans. Comm.* (June 1999), 47(6):799–803.

[45] Jiang, W., and A. Ortega. Multiple description coding via polyphase transform and selective quantization. In *SPIE Conf. Image Proc. Visual Comm. (VCIP'99)*, SPIE-3653(1–2):998–1008. San Jose, Jan. 1999.

[46] Kansari, M., et al. Low bit rate video transmission over fading channels for wireless microcellular systems. *IEEE Trans. Circuits Syst. for Video Technology* (Feb. 1996), 6:1–11.

[47] Katsaggelos, A. K., and N. P. Galatsanos, eds. *Signal Recovery Techniques for Image and Video Compression and Transmission.* Boston: Kluwer Academic Publishers, 1998.

[48] Kawahara, T., and S. Adachi. Video transmission technology with effective error protection and tough synchronization for wireless channels. In *IEEE Int. Conf. Image Proc. (ICIP'96)*, 101–04, Lausanne, Switzerland, Nov. 1996.

[49] Kurtenbach, J., and P. A. Wintz. Quantizing for noisy channels. *IEEE Trans. Commun.* (April 1969), 17:291–302.

[50] Lam, W.-M., A. R. Reibman, and B. Liu. Recovery of lost or erroneously received motion vectors. In *IEEE Int. Conf. Acoustics, Speech, Signal Proc. (ICASSP'93)*, Minneapolis, April 1993, 5:417–20.

[51] Lee, S. H., P. J. Lee, and R. Ansari. Cell loss detection and recovery in variable rate video. In *3rd Int. Workshop on Packet Video*, Morriston, NJ, March 1990.

[52] Lin, S., and D. J. Costello. *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice Hall, 1983.

[53] Marasli, R., P. D. Amer, and P. T. Conrad. Retransmission-based partially reliable transport service: An analytic model. In *IEEE INFOCOM'96*, 2:621–29, San Francisco, March 1996.

[54] Maxwell, K. Asymmetric digital subscriber line: interim technology for the next forty years. *IEEE Commun. Mag.* (Oct. 1996), 100–06.

[55] Mead, D. C. *Direct Broadcast Satellite Communications: An MPEG Enabled*

Service. Englewood Cliffs, NJ: Prentice Hall, 2000.

[56] Modestino, J. W., and D. G. Daut. Combined source-channel coding of images. *IEEE Trans. Commun.* (Nov. 1979), 7:1644–59.

[57] Modestino, J. W., D. G. Daut, and A. L. Vickers. Combined source-channel coding of images using the block cosine transform. *IEEE Trans. Commun.* (Sept. 1981), 29:1261–73.

[58] Ortega, A., and K. Ramchandran. Rate-distortion methods for image and video compression. *IEEE Signal Processing Magazine* (Nov. 1998), 15:23–50.

[59] Ramamurthy, G., and D. Raychaudhuri. Performance of packet video with combined error recovery and concealment. In *IEEE INFOCOMM'95*, 753–61, 1995.

[60] Redmill, D. W., and N. G. Kingsbury. The EREC: An error resilient technique for coding variable-length blocks of data. *IEEE Trans. Image Proc.* (April 1996), 5(4):565–74.

[61] Reibman, A., H. Jafarkhani, Y. Wang, and M. Orchard. Multiple description coding for video using motion compensated prediction. In *IEEE Int. Conf. Image Proc. (ICIP'99)*, 3:837–41, Kobe, Japan, Oct. 1999.

[62] Schulzrinne, H. IP networks. In M. T. Sun and A. R. Reibman, eds., *Compressed Video over Networks*. New York: Marcel Dekker, 2001, 81–138.

[63] Schulzrinne, H., S. Casner, R. Frederick, and V. Jacobson. *RTP: A transport protocol for real time applications*. IETF RFC 1889, Jan. 1996. (Available from ftp://ftp.isi.edu/in-notes/rfc1889.txt).

[64] Smith, B. C. Implementation Techniques for Continuous Media Systems and Applications. Unpublished Ph.D. thesis, University of California, Berkeley, 1994.

[65] Srinivasan, M., and R. Chellappa. Multiple description subband coding. In *IEEE Int. Conf. Image Proc. (ICIP'98)*, 1:684–88. Chicago, IL, Oct. 1998.

[66] Stallings, W. *ISDN and Broadband ISDN*. 2d ed. New York:Macmillan, 1992.

[67] Stuhlmuller, K., N. Färber, M. Link, and B. Girod. Analysis of video transmission over lossy channels. *IEEE Journal on Selected Areas in Communications* (June 2000), 18(6):996–1011.

[68] Sun, H., K. Challapali, and J. Zdepski. Error concealment in digital simulcast AD-HDTV decoder. *IEEE Trans. Consumer Electronics* (Aug. 1992), 38(3):108–17.

[69] Sun, H., and W. Kwok. Concealment of damaged block transform coded images using projections onto convex sets. *IEEE Trans. Image Process.* (April 1995), 4(4):470–77.

[70] Sun, H., and J. Zdepski. Error concealment strategy for picture-header loss in MPEG compressed video. In *SPIE Conf. High-Speed Networking and Multimedia Computing*, 145–52, San Jose, CA, Feb. 1994.

[71] Swann, R., and N. G. Kingsbury. Transcoding of MPEG-2 for enhanced resilience to transmission errors. In *IEEE Int. Conf. Image Proc. (ICIP'96)*, 2:813–16. Lausanne, Switzerland, Nov. 1996.

[72] Takishima, Y., M. Wada, and H. Murakami. Reversible variable length codes. *IEEE Trans. Commun.* (Feb. 1995), 43(2):158–62.

[73] Turletti, T., T. Huitema, and C. Huitema. *RTP payload format for H.261 video streams*, IETF RFC2032, Oct. 1996.

[74] Vaishampayan. V. A., Design of multiple description scalar quantizers. *IEEE Trans. Inform. Theory* (May 1993), 39:821–34.

[75] Vaishampayan, V. A., and N. Farvardin. Optimal block cosine transform image coding for noisy channels. *IEEE Trans. Commun.* (March 1990), 38:327–36.

[76] Villasenor, J. D., Y.-Q. Zhang, and J. Wen. Robust video coding algorithms and systems. *Proceedings of the IEEE* (Oct. 1999), 87:1724–33.

[77] Wada, M. Selective recovery of video packet loss using error concealment. *IEEE J. Select. Areas Commun.* (June 1989). 7(5):807–14.

[78] Wang, M., and T. R. Fischer. Trellis coded quantization designed for noisy channels. *IEEE Trans. Inform. Theory* (Nov. 1994), 40:1792–802.

[79] Wang, Y., and D. Chung. Non-hierarchical signal decomposition and maximally smooth reconstruction for wireless video transmission. In D. J. Goodman and D. Raychaudhuri, eds., *Mobile Multimedia Communications*. New York: Plenum Press, 1997, 285–92.

[80] Wang, Y., M. Orchard, V. Vaishampayan, and A. R. Reibman. Multiple description coding using pairwise correlating transform. *IEEE Trans. Image Process.* (March 2001), 10(3):351–66.

[81] Wang, Y., S. Wenger, J. Wen, and A. G. Katsaggelos. Error resilient video coding techniques. *IEEE Signal Processing Magazine* (July 2000), 17(4):61–82.

[82] Wang, Y., and Q.-F. Zhu. Error control and concealment for video communication: A review. *Proceedings of the IEEE* (May 1998), 86:974–97.

[83] Wang, Y., and Q.-F. Zhu. Maximally smooth image recovery in transform coding. *IEEE Trans. Commun.* (Oct. 1993), 41(10):1544–51.

[84] Wen, J., and J. Villasenor. Utilizing soft information in decoding of variable length codes. In *Data Compression Conference (DCC'99)*, Snowbird, UT, March 1999, 131–39.

[85] Wen, J., and J. Villasenor. Reversible variable length codes for robust image and video transmission. In *Data Compression Conference (DCC'98)*, Snowbird, UT, April 1998, 471–80.

[86] Wen, J., and J. Villasenor. A class of reversible variable length codes for robust image and video coding. In *IEEE Int. Conf. on Image Proc. (ICIP'97)*, 2:25–28. Santa Barbara, CA, Oct. 1997.

[87] Wenger, S., Video redundancy coding in H.263+. In *Workshop on Audio-Visual Services for Packet Networks (AVSPN'97)*, Aberdeen, Scotland, Sept. 1997.

[88] Wenger, S., G. Knorr, J. Ott, and F. Kossentini. Error resilience support in H.263+. *IEEE Trans. Circuits Syst. for Video Technology* (Nov. 1998), 8(6):867–77.

[89] Wu, D., et al. An end-to-end approach for optimal mode selection in Internet video communication: Theory and application. *IEEE Journal on Selected Areas in Communications* (June 2000), 18(6):977–95.

[90] Wu, D., et al. On end-to-end architecture for transporting MPEG-4 video over the Internet. *IEEE Trans. Circuits Syst. for Video Technology* (June 2000), 18(6):977–95.

[91] Yang, X., and K. Ramchandran. Optimal multiple description subband coding. In *IEEE Int. Conf. Image Proc. (ICIP'98)*, 1:684–58, Chicago, IL, Oct. 1998.

[92] Yu, G.-S., M. M.-K. Liu, and M. W. Marcellin. POCS-based error concealment for packet video using multiframe overlap information. *IEEE Trans. Circuits Syst. for Video Technology* (Aug. 1998), 8(4):422–34.

[93] Zeger, K., and A. Gersho. Pseudo-Gray coding. *IEEE Trans. Commun.* (Dec. 1990), 38:2147–56.

[94] Zhang, R., S. L. Regunathan, and K. Rose. Video coding with optimal inter/intra-mode switching for packet loss resilience. *IEEE Journal on Selected Areas in Communications* (June 2000), 18(6):966–76.

[95] Zhang, Y.-Q., Y. J. Liu, and R. L. Pickholtz. Layered image transmission over cellular radio channels. *IEEE Trans. Vehicular Tech.* (Aug. 1994), 43:786–96.

[96] Zheng, H., and K. J. R. Liu. Multimedia services over digital subscriber lines. *IEEE Signal Process. Mag.* (July 2000), 17(4):44–60.

[97] Zhu, Q.-F. Device and method of signal loss recovery for real-time and/or interactive communications. U.S. Patent 5,550,847, Aug. 1996.

[98] Zhu, Q.-F., V. Eyuboglu, and M. Sridhar. Device and method of digital video streaming, U.S. Patent 5,768,527, June 1998.

[99] Zhu, Q.-F., and Y. Wang. Error concealment in visual communications. In M. T. Sun and A. R. Reibman, eds., *Compressed Video over Networks*. New York: Marcel Dekker, 2001, 217–50.

[100] Zhu, Q.-F., Y. Wang, and L. Shaw. Coding and cell loss recovery for DCT-based packet video. *IEEE Trans. Circuits Syst. for Video Technology* (June 1993), 3(3):248–58.

第 15 章　因特网和无线 IP 网络上的流视频

当前计算技术、压缩技术、宽带存储设备和高速网络的发展使得通过因特网提供实时多媒体服务成为可能。正如其名称所隐含的，实时多媒体具有时间上的限制。例如，视频和音频数据必须不间断地播放，如果数据不能按时到达，播放过程将会暂停，这是令观众讨厌的。

对现场或存储的视频进行实时传输是实时多媒体的主要部分。在本章中我们所讨论的视频流是指存储视频的实时传输①。视频流一般有带宽、延时和损耗的要求。然而，目前最尽力的因特网不对流视频提供任何 QoS 保证。此外，因特网的非均匀性使它难于支持在视频组播中灵活地提供服务，满足用户宽范围的 QoS 需求。更有甚者，对于无线 IP 网络上的流视频，无线信道条件的无规则波动会造成视频质量的严重退化。因此，在因特网和无线 IP 网络上的流视频面临着许多挑战。

为了对付这些挑战，人们已经付出了大量的努力。以提供这一领域的一个概览为目的，我们囊括了关于流视频的 7 个领域，分别称为：视频压缩、流视频的应用层 QoS 控制、连续媒体发布服务、流服务器、媒体同步机制、流媒体协议和无线 IP 网络上的流视频。对每个领域，我们将讨论特定的问题并考察有代表性的方法和体系。

15.1　视频流系统的结构

一个视频流系统通常包括 7 个组成部件，如图 15.1 所示。在图中，原始的视频和音频数据由视频和音频压缩算法进行预压缩，然后存储在存储设备中。依据用户的请求，流服务器从存储设备中检索到压缩的视频/音频数据，然后应用层 QoS 控制模块根据网络状况和 QoS 要求调节视频/音频比特流。调节后，传输协议对压缩的比特流进行打包，并把视/音频数据包送到因特网或无线 IP 网络。在因特网上，由于拥塞一些数据包可能会丢失或经历过长的延迟；在无线 IP 段，有些包可能因误码而遭到破坏。为了提高视/音频的传输质量，连续媒体发布服务被用于因特网。那些成功地到达接收端的数据包首先经过传输层，然后经应用层处理，再由视/音频解码器解码。为了达到视频和音频演播的同步，需要媒体同步机制。由图 15.1 可以看到，这 7 个部分是紧密联系的，它们是视频流结构的连贯的要素。

下面，我们分别简要地介绍这 7 个部分。

1. **视频压缩**：为了实现高效的传输，原始视频在传输之前必须进行压缩。视频压缩方案可以分为两类：可分级和不可分级的视频编码。由于可分级视频能够得体地应付因特网的带宽波动[45]，所以我们主要涉及可分级视频编码技术。在第 11 章中已经讨论过这个课题，因此这里只给出这个课题的一个简单的回顾。
2. **流视频应用层 QoS 控制**：为了适应变化的网络条件和用户对演播质量变化的要求，已

① 视频流指的是视频内容不需要全部下载，而是在部分内容被接收和解码时进行重放。

经提出了各种应用层 QoS 控制技术[19,69,77]。这些技术包括拥塞控制和差错控制。它们各自的功能如下:拥塞控制用于防止丢包和减小延迟;差错控制则是在出现丢包的情况下改善视频演播的质量。一般视频应用中的差错控制已经在第 14 章做过介绍,这里,我们将集中讨论在视频流应用中有效的那些差错控制技术。

3. **连续媒体发布服务**:要提供有质量的多媒体演播,网络的支持是重要的,这是因为网络支持可以减小传输延迟和丢包率。由于建立在因特网顶层(IP 协议),连续媒体发布服务对于最尽力的因特网上的流视频能够实现 QoS 和高效传输。连续媒体发布服务包括网络滤波、应用层组播和内容复制。

4. **流服务器**:流服务器在提供流服务中起着关键的作用。为了提供有质量的流服务,要求流服务器在一定时限内处理多媒体数据,并且支持交互式控制操作,如暂停/继续演播、快进和快退。此外,流服务器必须以同步方式重新得到多媒体内容。一个流服务器典型地包含三个子系统,它们是通信程序(即传输协议)、操作系统和存储系统。

5. **媒体同步机制**:媒体同步机制是多媒体应用区别于其他传统数据应用的一个主要特征。在媒体同步机制下,接收端的应用可以按与各媒体原始采集相同的方式提供各种媒体流。媒体同步的一个典型的效果是说话者嘴唇的运动与演播的音频相一致。

6. **流媒体协议**:协议是为客户与流服务器之间的通信而设计和标准化的。流媒体协议提供的服务包括网络寻址、传输和话路控制等。协议可分为三类:(1)网络层协议,如 IP,(2)传输协议,如用户数据包协议(UDP),(3)话路控制协议,如实时流协议(RTSP)。

7. **无线 IP 网络上的流视频**:无线信道的不稳定状态使无线 IP 网络上提供 QoS 视频传输面临着许多挑战。为解决这个问题,提出了自适应帧结构[85]。自适应帧结构包括可分级的视频表示、网络感知视频应用和自适应服务。

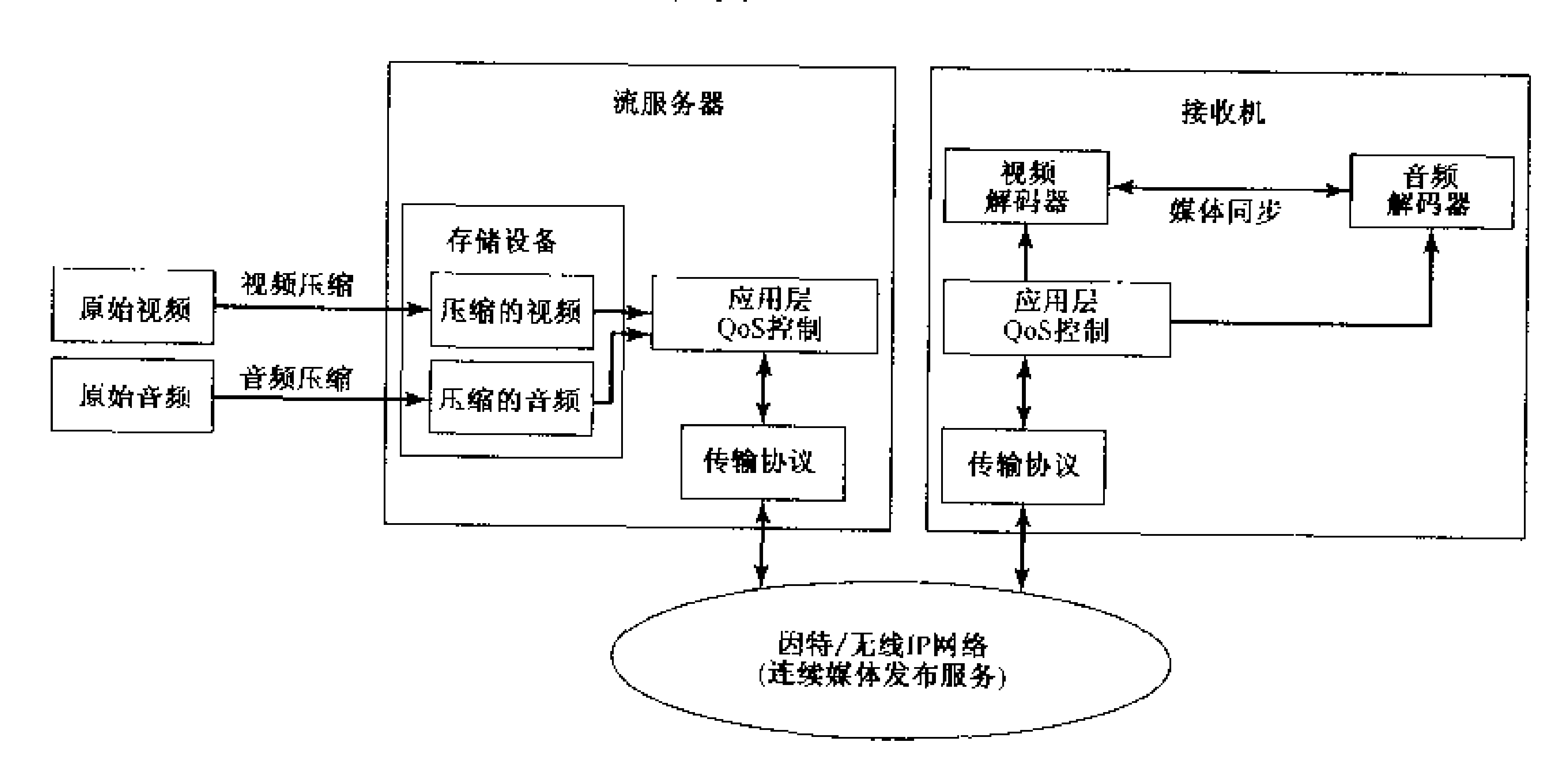


图 15.1 视频流的结构

本章的其他部分是这样安排的:15.2 节讨论视频压缩技术。在 15.3 节,我们给出流视频的应用层 QoS 控制机制。15.4 节描述连续媒体发布服务。在 15.5 节,讨论设计流服务器的关键问题。15.6 节介绍各种媒体同步机制。15.7 节综述流视频的关键协议。15.8 节介绍一种

无线 IP 网络上流视频的自适应帧结构，最后，在 15.9 节中给出本章的小结。

15.2 视频压缩

由于原始视频占用大量的带宽，在传输之前必须进行压缩以便实现高效传输。视频压缩方案可以分为两类：可分级的和不可分级的视频编码。如第 11 章所述，可分级视频编码方案产生压缩的比特流，它的一部分是可解码的。与解码全部的比特流相比，解码部分的压缩比特流将产生降质的图像，或者是较小的图像尺寸，或者是较小的帧率。已经证明，可分级视频能够很好地适应因特网带宽的波动[40,45]。相反，不可分级视频容易受带宽变化的影响，因为它不能使视频表示适应带宽的变化[45]。而且，可分级视频表示是在异类网络(例如具有不同接入带宽的网络)上实现组播的一种高效、灵活的方法[40,45]。由于这些原因，所有的流视频服务都使用可分级的视频编码技术。关于各种可分级编码方法的讨论见第 11 章。下面介绍应用层 QoS 控制机制，它能够根据网络状况和 QoS 需求调节视频比特流。

15.3 流视频的应用层 QoS 控制

在出现丢包和可用带宽变化的情况下，应用层 QoS 控制可以最大限度地提高视频质量。应用层 QoS 控制技术包括拥塞控制和差错控制。这些技术应用于终端系统，并不需要路由器和网络的 QoS 支持。15.3.1 节概述拥塞控制方法，15.3.2 节描述差错控制的机制。

15.3.1 拥塞控制

突发丢失和超时延迟对视频演播质量具有破坏性的效果，而它们往往是由于网络拥塞造成的。减少丢包和延迟的一个方法是在信源和/或接收端应用拥塞控制。

拥塞控制有两种机制：码率控制和码率整形。码率控制通过使视频流的码率与可用的网络带宽相匹配，力求将网络拥塞和丢包的数目降至最低。而码率整形迫使信源以码率控制算法规定的码率发送视频流。

码率控制 码率控制是一个基于估计的网络中的可用带宽确定视频通信发送码率的技术[①]。现有的码率控制方案可分为三类：基于信源的、基于接收端的和混合的码率控制。

基于信源的码率控制：在基于信源的码率控制下，由发信方负责调节视频传输码率。通常，基于信源的码率控制机制要利用反馈。发信方根据网络的反馈信息调节视频流的码率。基于信源的码率控制可应用于单播[81]和组播[8]。

对于单播，现有的基于信源的码率控制机制按照两种方法实现：基于探测器的方法和基于模型的方法。

基于探测器的方法建立在探测实验的基础上。明确地说，信源通过调节发送的码率探测可用的网络带宽，方法是使丢包率 p 小于某个阈值 P_{th}[81]。有两种方法用于调节发送码率：

① 在 9.3.4 节我们把码率控制问题定义为(1)确定适当的编码速率，(2)调节编码参数以满足目标码率。本节中的码率控制只是指第一项任务。

(1)加法提高,乘法降低[81],(2)乘法提高,乘法降低[74]。

基于模型的方法建立在一个 TCP 连接的吞吐量模型基础上。明确地说,一个 TCP 连接的吞吐量可由下式表征:

$$\lambda = \frac{1.22 \times \mathrm{MTU}}{\mathrm{RTT} \times \sqrt{p}} \tag{15.3.1}$$

其中 λ 是 TCP 连接的吞吐量,MTU(最大传输单元)是该连接所用的包的大小,RTT 是这个连接的往返时间,p 为该连结经历的丢包率。在基于模型的码率控制下,公式(15.3.1)用于决定视频流的发送码率。这样,视频连接可以通过类似于 TCP 的方式避免拥塞,并且可以与 TCP 流公平竞争。因此,基于模型的码率控制又称为"TCP 友好的"码率控制[22]。

对于基于信源的码率控制下的组播,发信方使用一条通道把视频发送到接收端。所以组播又称为"单信道组播"。对于单通道组播只能使用基于探测器的码率控制[8]。

由于所有的接收端共用一条信道,单信道组播是高效的。然而单信道组播不能提供灵活的服务以满足具有各种接入带宽的接收端的不同需求。相反,如果通过各自的单播流传送多点视频,那么带宽的利用率很低,但服务可以被区分开,因为每个接收端都可以与信源协商服务参数。单播和单通道组播是两种极端的情况,如图 15.2 所示。为了实现多点传送视频在带宽利用率和服务灵活性之间较好的折中,已经提出了基于接收端的码率控制和混合的码率控制。

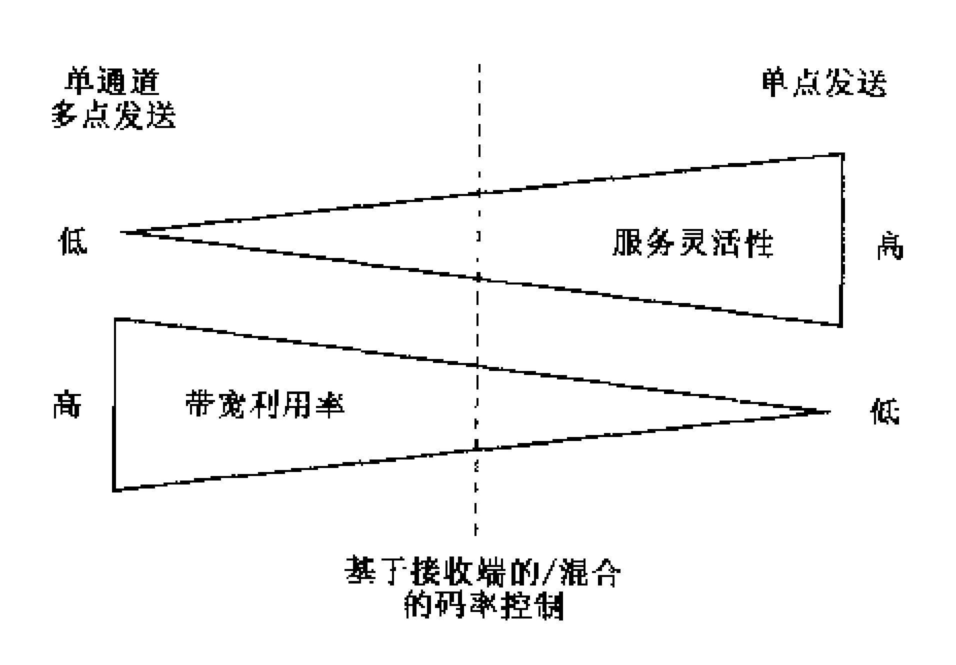


图 15.2 利用率和灵活性之间的折中

基于接收端的码率控制:在基于接收端的码率控制下,接收端通过增加或丢弃通道来调节接收的视频流码率,而发信方不参加码率控制。一般,基于接收端的码率控制只应用于分层的组播,而不应用于单播视频。

类似于基于信源的码率控制,现有的基于接收端的码率控制机制按照两种方法实现:基于探测器的方法和基于模型的方法。基本的基于探测器的码率控制[45]包含两部分:

1. 当没有检测到拥塞时,接收端通过增加一层来探测可利用带宽,从而提高了它的接收码率。若在加入一层后没有检测到拥塞,则加入实验成功。否则,接收端丢掉新加入的层。
2. 如果检测到拥塞,接收端丢掉一层,使它的接收码率降低。

不同于通过探测实验隐含地估计可用网络带宽的基于探测器的方法，基于模型的方法直接估计可用网络带宽。基于模型的方法是基于 TCP 连接的吞吐量模型，即公式(15.3.1)，如在基于信源的码率控制下所描述的那样。因此基于模型的码率控制也是“TCP 友好的”。

混合的码率控制：在混合的码率控制下，接收端通过增加或丢弃通道来调节接收码率，同时发信方根据从接收端反馈回的信息也调节发送码率。混合码率控制的一个例子是目的地集分组[13]。

码率整形　通过码率整形技术可以将预压缩的视频比特流的码率修正到目标码率限制。码率整形器是压缩层与网络传输层之间或两个网络段之间的一个接口(或滤波器)，通过它可以达到视频流与可用带宽之间的匹配。

有许多类型的码率整形器或滤波器[87]，包括：

1. **编解码滤波器**：编解码滤波器对视频流进行压缩和解压。它们通常用于实现不同压缩方案之间的转码。取决于所用的压缩算法，转码可以简化，无须全部解压缩和再压缩。
2. **弃帧滤波器**：弃帧滤波器可以辨别帧的类型(例如 MPEG 中的 I 帧、P 帧和 B 帧)，并根据帧的重要程度丢弃某些帧。例如，弃帧顺序可能首先是 B 帧，其次是 P 帧，最后是 I 帧。弃帧滤波器通过丢掉一定数目的帧降低视频流的码率。剩余的帧只需要较低的码率。这种滤波器可以用于信源[90]或网络上(见 15.4.1 节)。
3. **弃层滤波器**：弃层滤波器可以识别并根据重要程度丢弃某些层。丢弃顺序从最高增强层往下到基本层。
4. **频率滤波器**：频率滤波器在压缩层进行运算。明确地说，它们工作在频域(即 DCT 系数)。频率滤波机制包括低通、退色和彩色到单色滤波。

   低通滤波器丢掉较高频率的 DCT 系数。除了只处理视频流中的色度信息以外，退色滤波器的作用类似于低通滤波器。彩色到单色滤波器从视频流中去掉全部的彩色信息。在 MPEG 中，这是通过将色度块用空块替换实现的。

   与弃帧滤波器不同，频率滤波器在降低带宽的同时不影响帧率，而是降低了最终帧的显示质量。
5. **再量化滤波器**：再量化滤波器在压缩层进行运算(即 DCT 系数)。这种滤波器首先通过反量化从压缩的视频流中提取 DCT 系数，然后用较大的量化步长对 DCT 系数进行量化，从而降低了码率。

综上所述，拥塞控制的目的是防止丢包。然而事实是在因特网上丢包是不可避免的，并且可能显著地影响接收质量，这就促使了在出现丢包的情况下使视频演播质量最优化的机制的设计。差错控制就是这样一种机制。

15.3.2 差错控制

在第 14 章讨论过视频通信中的差错控制技术。这里，我们把差错控制技术分为四类：传输层差错控制，包括 FEC 和限制延迟重发；编码器差错复原编码；解码器错误隐藏；和编码器解码器交互的差错控制。这里我们描述几种在视频流应用中有效的差错控制技术。

FEC　FEC 靠在原始消息流中加入冗余信息工作，使得在出现丢包的情况下可以重建消

息。在因特网应用中,一般是穿越包进行块编码。具体地说,首先将视频流分成段,每一段打成 k 个包;然后对每一段穿越这 k 个包进行块编码(例如 Tornado 编码[1]),生成一个具有 n 个包的块,其中 $n > k$。为完全恢复一个段,用户只需在具有 n 个包的块中接收到任意 k 个包。

限制延迟重发 通常在实时视频传输中不考虑重发的方法,因为重发的包可能错过它的演播时间。然而,在流服务中,如果包的单程传输时间小于最大的允许延迟,那么限制延迟重发是差错控制的一个可行的选择(见 14.3.3 节)。

对于单播,根据由谁决定向何处发送和/或响应重发请求,已经提出了三种限制延迟的重发机制:基于收信方的、基于发信方的和混合的控制。

基于收信方控制的目标是使那些不能按时到达并显示的重发请求最少。基于收信方的控制下,收信方执行如下算法:

如果收信方检测到 N 个包丢失:
  若($T_c + \text{RTT} + D_s < T_d(N)$)
    则向发信方送出 N 个包的请求

其中 T_c 是当前的时刻,RTT 是估计的往返时间,D_s 是松弛项,$T_d(N)$是包 N 显示的确定时刻。松弛项 D_s 可以包括 RTT 的估计误差容限、发信方的响应时间和收信方的解码延时。基于收信方控制的定时图如图 15.3 所示,其中 D_s 只是收信方的解码延时。

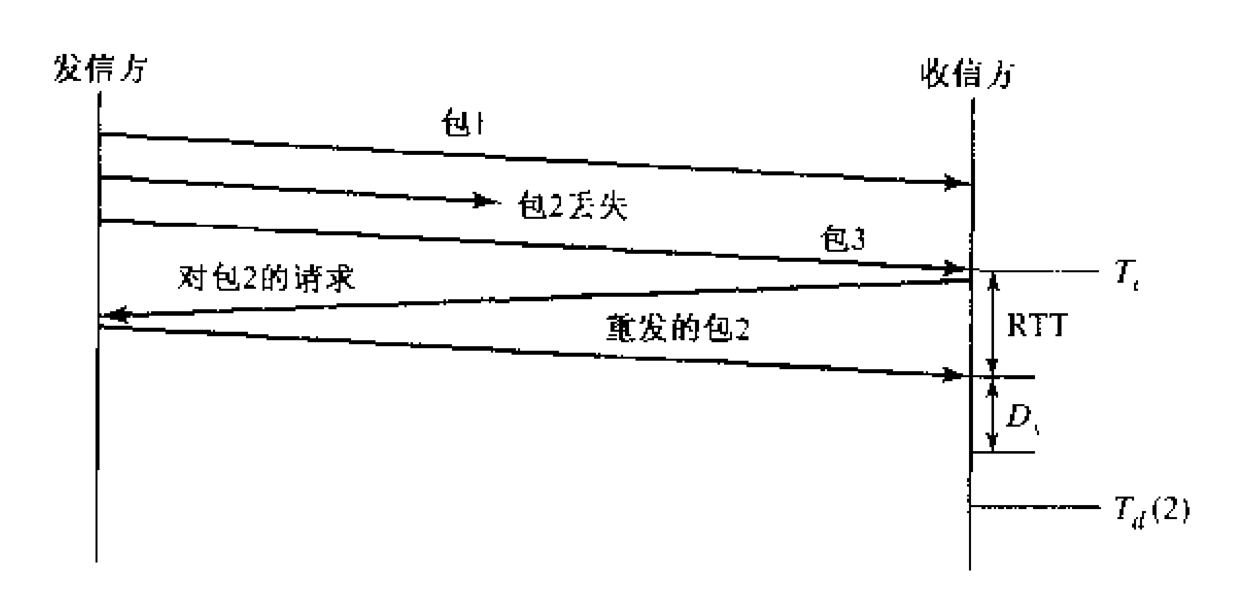


图 15.3 基于收信方控制的定时图

基于发信方控制的目标是取消那些将错过在收信方的显示时刻的重发请求。在基于发信方的控制下,发信方执行如下算法:

当发信方收到 N 个包的请求时:
  若($T_c + \text{RTT}/2 + D_s < T'_d(N)$)
    则重发 N 个包到收信方

其中 $T_d'(N)$是 $T_d(N)$的估计值。

混合控制是基于发信方和基于收信方控制的简单联合。

在组播的情况下,重发必须限制在位置较近的组播成员之内,这是因为在这些成员之间单程传输时间会比较小,使得重发的效率较高。典型地是配置一个逻辑树来限制重发需求的数目和范围,并且实现位置较近的组播成员之间的差错复原[86]。此外,为解决异类问题,可以

采用一个差错复原的收信方启动机制[86]。

差错复原编码　如 14.4 节所述,差错复原技术是从压缩的角度解决丢失复原问题。具体地说,它们力图阻止误差累积或限制在压缩层中破坏(由丢包引起的)的范围。标准化的差错复原工具包括重同步标记、数据划分和可逆的可变长编码[35,68](见 14.4 节)。这些技术主要针对比特错误占主导地位的环境,如无线通道。对于因特网上的视频应用,由于采用基于包的传送方式,一个丢包有可能导致所有运动数据以及相关的形状和纹理数据的丢失,这些技术可能不是很有用的。而且,包的边界已经为收信方提供了可变长编码比特流的同步点,使得重同步标记有些多余。

对于健壮的因特网视频传输,更有希望的两种技术是最佳模式选择和多描述编码。

最佳帧内/帧间模式选择:丢包对于视频演播质量的影响依赖于信源采用的编码方案、网络的拥塞状况以及接收端采用的错误隐藏方案。高压缩率编码算法通常采用帧间编码(即预测)以获得编码效率。对于这些编码算法,一个包的丢失有可能导致大量帧的视频质量退化,直到接收到下一个帧内编码帧为止。帧内编码可以有效地阻止误差累积,代价是编码效率不高;而帧间编码可以提高压缩效率,风险是误差累积。因此一个好的在帧内模式和帧间模式之间的选择机制应该增强由帧内和帧间编码所压缩的视频的健壮性(见图 15.4)。

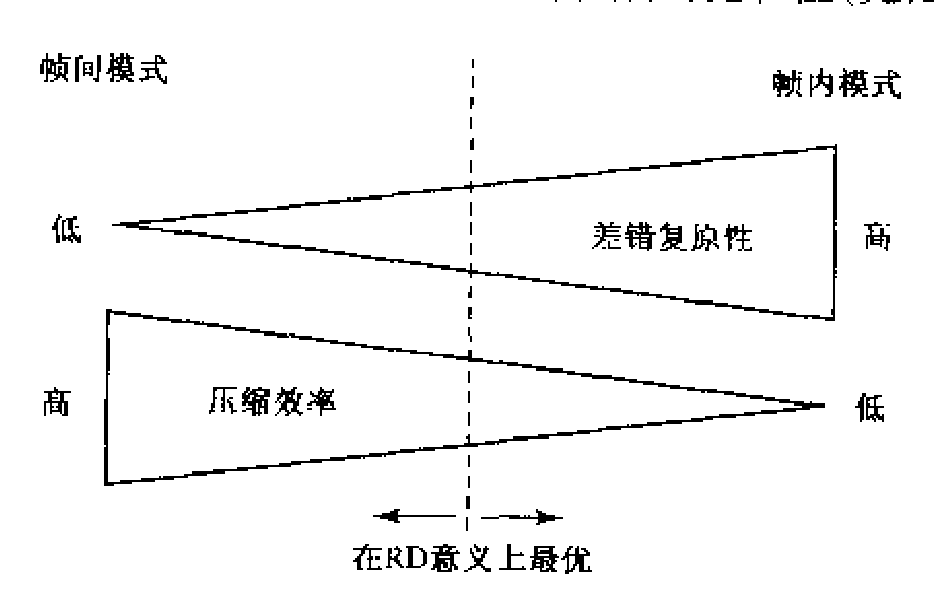


图 15.4　最佳模式选择取决于网络条件

对于网络视频传输,如 H.263 或 MPEG-4[35]等编码算法通常会调节输出码率,使之与可用带宽相匹配。调节码率的压缩算法的目标是在给定比特预算的限制下,使视频质量最优。这可以通过在给定的比特预算下,选择一种使原始与重建帧或宏块之间的量化失真为最小的模式来实现[52],这就是所谓的 RD 最优模式选择(见 9.3.3 节)。我们把这种 RD 最优模式选择称为经典方法,经典方法在易出错的环境下不能实现全局最优,因为它没有考虑网络拥塞的状况和接收端的特性。

为解决这个问题,Wu 等[83]为 RD 最优模式选择提出了一种端对端的方法。这个方法考虑了对于接收端视频演播质量具有影响的所有三个因素:(1)信源的特点;(2)信道的特性;(3)接收端的特点。由于考虑了网络拥塞状态和接收端的特点,已经证明这种方法对于因特网视频应用能够给出比经典方法更优越的性能[83]。其他的考虑了传输差错效应的模式选择方法已经在 14.4.3 节介绍了。

多描述编码:多描述编码(14.4.5 节)是另一种兼顾压缩效率和丢包健壮性的方法。与 14.4 节所述的其他差错复原技术相比,它更能有效地解决丢包问题。为了在因特网上传送

MDC流，来自不同描述的比特应该分别打包，并最好通过不同的路径传输。另一种方法是，把含有不同描述的包适当地进行交织，使得通信拥塞不会导致丢失所有的描述。

错误隐藏 当检测到丢包时，接收端可以利用错误隐藏来隐藏丢失的数据，从而使显示更加悦目。由于观众可以容忍一定程度的视频信号失真，错误隐藏是解决丢包问题的一种可行的技术[80]。

如14.5节所述，错误隐藏有两种基本的方法，称为空间内插和时间内插。在空间内插中，丢失的像素值由相邻的空间信息重建；而在时间内插中，丢失的数据由以前帧的数据重建。一般情况下，空间内插用于重建帧内编码帧中丢失的数据，而时间内插用于重建帧间编码帧中丢失的数据。

对于需要实时解码的流应用，14.5节中介绍的大多数技术过于复杂。最可行和有效的技术是运动补偿时间内插(见14.5.1节)，在这种方法中，接收端用前一帧中由运动矢量指示的块代替受损的块。如果邻近的运动矢量可以利用，这个运动矢量可以从受损块邻块复制，否则就将运动矢量置为零。

以上我们回顾了各种应用层QoS控制技术。这些技术被终端系统所采用，并不需要路由器或网络的支持。若网络支持视频流的QoS，则性能将进一步增强。下面，我们给出为最尽力的因特网开发的QoS支持机制。

15.4 连续媒体发布服务

为了提供高质量的多媒体演播，网络的支持是重要的。这是因为网络的支持可以降低传输延迟和丢包率。流视频和音频被归类为连续媒体，因为它们是由媒体量子(如音频采样或视频帧)序列组成的，这些媒体量子只当按时演播时才有传送的意义。由于建立在因特网顶层(IP协议)，连续媒体发布服务设计的目的是提供QoS，并且实现流视频和音频在最尽力的因特网上的高效传输。连续媒体发布服务包括网络滤波、应用层组播和内容复制，这些将分别在15.4.1节~15.4.3节中介绍。

15.4.1 网络滤波

类似于拥塞控制技术，网络滤波的目的也是在网络拥塞时使视频质量最优化。如15.3.1节所述，视频服务器中的滤波器可以根据网络拥塞的状况调节视频流的码率。然而，要视频服务器处理为调节每一个单播视频流所需要的计算，可能是过于繁忙了。因此服务提供者希望把滤波器配置在网络上[32]。图15.5示出了一个网络滤波的例子。在图中，节点"R"代表路由器，它并不知道媒体流的格式，并且会随机地丢弃数据包。"滤波器"节点接收客户的请求，并相应地调节从服务器送出的码流。这个解决方案允许服务提供者将滤波器配置在连接网络瓶颈的节点上，并且服务器到客户的路径上可以有多个滤波器。

为了说明滤波器的作用，一个系统模型示于图15.6[32]。这个模型包括服务器、客户、至少一个滤波器以及它们之间的两条虚拟通道。这两条虚拟通道，一条用于控制，另一条用于数据。同样的通道存在于任意一对滤波器之间。控制通道是双向的，可由TCP连接实现。图15.6所示的模型允许客户只与一个主机(最近的滤波器)通信，主机或者将请求上传或者响应这些请求。滤波器在数据面上的操作包括：(1) 接收来自服务器或上一级滤波器的视频流；

(2)把视频数据以目标码率发送给客户或下一级滤波器。滤波器在控制面上的操作包括:(1)接收来自客户或下一级滤波器的请求;(2)响应请求;(3)将请求传至上一级滤波器。

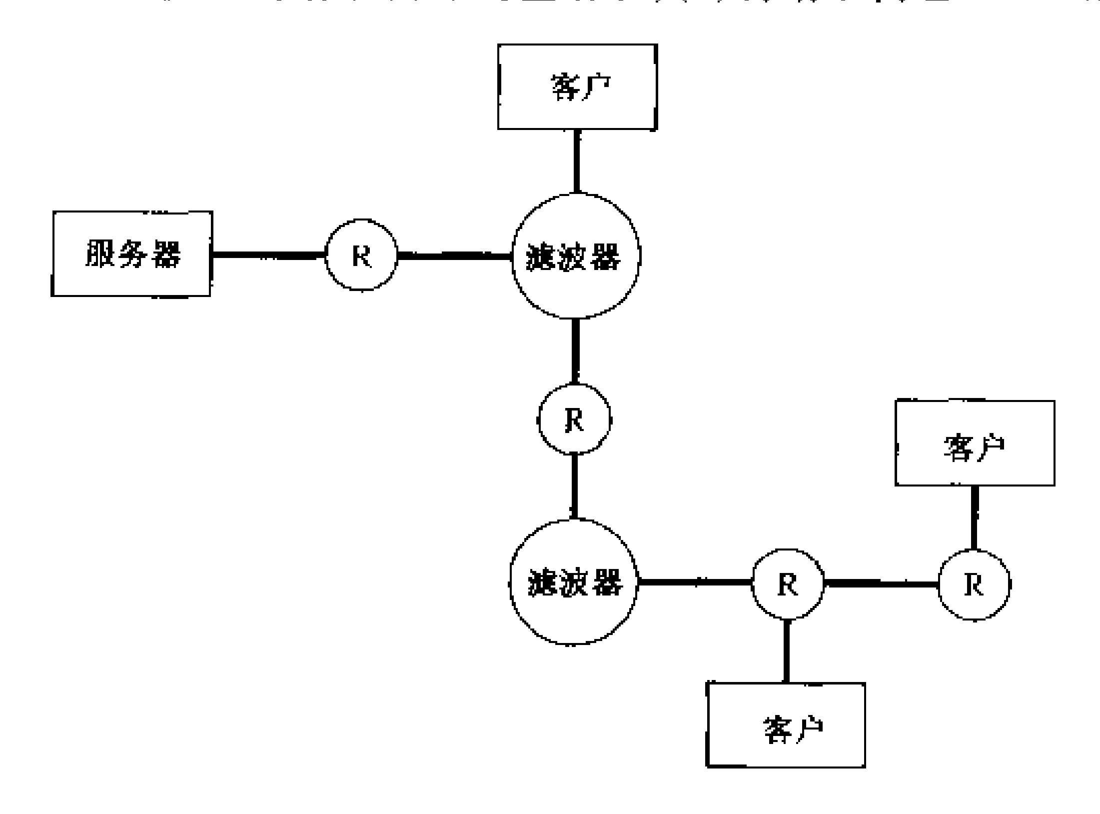


图 15.5　网络滤波器

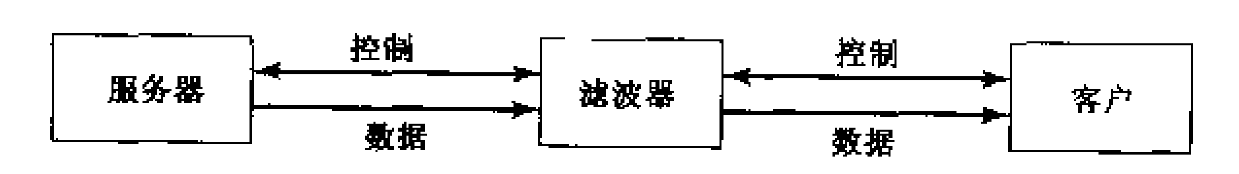


图 15.6　一个网络滤波的系统模型

弃帧滤波器(见 15.3.1 节)一般作为网络滤波器使用。接收端通过向滤波器发送增加或减小弃帧率的请求,可以改变媒体流的带宽。为了便于确定滤波器是应该增加还是应该减小带宽,接收端不断地测量丢包率 p。根据丢包率,码率整形机制按如下方法设计[32]。如果丢包率大于阈值 α,则客户将请求滤波器增加弃帧率;如果丢包率小于另一个阈值 $\beta(\beta<\alpha)$,接收端则请求滤波器减小弃帧率。

在网络上使用弃帧滤波器的优点如下:

- **改善视频质量**　例如,当视频流从一个具有较大可用带宽的上游线路流向一个具有较小可用带宽的下游线路时,在线路的结合点使用一个弃帧滤波器将有助于改善视频质量。这是因为滤波器理解媒体流的格式,并且所采取的丢包方式是适度地降低流质量,而不是直接地破坏流。
- **带宽利用率高**　通过丢弃迟到的帧或因要素包的丢失而遭到破坏的帧,滤波可以节省网络资源。

注意,网络滤波针对的是单播(即点对点)媒体流。下面介绍一种流媒体的组播机制。

15.4.2　应用层组播

因特网的原始设计完全适于点对点的应用,如电子邮件、文件传输和 Web 浏览,但不能有效地支持大容量的传输,如流媒体组播。为了克服这个缺点,20 世纪 90 年代初提出了"IP 组播"技术。作为 IP 层的一个扩展,IP 组播能够提供高效的多点数据包传送。其特点是,传输的

有效性是通过保证沿IP组播树中任何一个物理路径只传输原始IP数据包(由组播信源送出的)的一个副本而实现的。然而,经过十年有价值的研究和开发,IP组播依然受到诸如可分级性、网络管理、网络配置和支持高层功能(如差错控制、流控制和拥塞控制)等有关问题的困扰。为解决这些问题,提出了应用层组播机制[21],应用层组播的目标是在因特网和传统IP网络的顶层建立组播服务。

应用层组播机制允许独立的内容服务供应商(CSP)、因特网服务供应商(ISP)或企业建立自己的组播网络,并将它们接入到更大的、世界范围的"媒体组播网络"。这就是说,媒体组播网络可以支持应用层或流媒体/内容层上的"对等关系",在那里,内容中枢链路与服务供应商互相连接。因此,正如因特网是由各ISP之间通过IP层的对等关系实现的网络相连建立起来的一样,媒体组播网络可以由多个内容发布网络互相连接建立起来,这些网络是通过各种服务供应商——如传统的ISP、CSP和应用服务供应商(ASP)——之间应用层的对等关系实现的。

媒体组播网络的工作描述如下。在媒体组播网络中,每一个能够组播的节点(称之为媒体桥[21])在应用层进行路由选择。此外,每一个媒体桥与一个或多个相邻的媒体桥通过显式配置相互连接,这种配置定义了应用层的重叠拓扑结构。一个媒体组播网络中的媒体桥都共同地使用一个分布式应用层组播路由算法,为通过网络传播内容确定最佳的虚拟路径。当所使用的网络发送失败或过于拥挤时,媒体组播网络根据应用层路由策略,自动地、动态地把内容改换到另一条路径上。此外,当且只当下游用户请求时媒体桥才动态地预定组播内容。这就保证了在任一条物理的或虚拟的路径上有且只有一个组播内容的副本流过,与下游用户的数量无关,从而节省了网络带宽。

应用层组播的优越性在于它打破了在可分级性、网络管理和支持拥塞控制之间的界限,而这个界限曾阻止各ISP建立"IP组播"的对等配置。

15.4.3 内容复制

提高媒体传送系统的可分级性的一个基本技术是内容/媒体复制。内容复制采取两种方式,称为缓冲存储和镜像存储,它们已被发行商、CSP和ISP所配置。不论是缓冲存储还是镜像存储,都是寻求将内容存放在距离客户最近的地方,且二者共有如下优点:

- 减小了网络线路上的带宽消耗;
- 减小了流服务器的负担;
- 减小了客户的等待时间;
- 提高了可用性。

镜像存储将原始多媒体文件的副本分散地存放在因特网的多个服务器上。也就是说,原始多媒体文件存放在主服务器上,同时这些文件的副本存放在复制服务器上。通过这种方法,客户就可以从离它最近的复制服务器上提取多媒体数据,从而为用户提供最好的性能(如最小的等待时间)。镜像存储具有若干优点,例如:

- 所有的内容都被复制下来;
- 发行商可以看到服务器访问日志,从而实现对用户的跟踪。

另一方面,镜像存储也有一些缺点。目前,建立专用的镜像存储的机制是昂贵的、特殊的和缓慢的。此外,在一个现有的(不是复制)服务器上建立镜像存储,即使是便宜的,依然是一

个特殊的和管理复杂的过程。最后,还没有标准的方法使文稿编排程序和服务器设置能容易地从一个服务器移植到另一个服务器。

缓冲存储是基于假定许多不同的客户会下载相同的内容,在本地制作内容的副本供客户取用。单一组织内的客户一般从单一的称为超高速缓冲存储器的本地机取出所有的内容。超高速缓冲存储器从流服务器读取一个视频文件,存储它的一个本地副本,然后发送给请求它的客户。如果客户所请求的文件已经存储在超高速缓冲存储器上,超高速缓冲存储器将直接把文件返送给客户,而不是再返回到放置该视频的流服务器。此外,缓存共享和缓存器体系允许每个超高速缓冲存储器访问其他超高速缓冲存储器上的文件,这样就减轻了主服务器的负担,缓解了系统瓶颈[11,20]。

缓冲存储有如下优点:

- 超高速缓冲存储器可以形成一个所有流服务器共享的内部结构;
- 超高速缓冲存储器的可配置性日益增加。

因为上述原因,ISP 已经成为缓冲存储的最大支持者。然而,超高速缓冲存储器不能提供发行商所需的服务:支持服务质量和安全性。首先,当超高速缓冲存储器存储了一个发行商的文件时,它不能保证这些文件的处理质量。发行商无法要求它的文件在特定的时限内传送,或者要求超高速缓冲存储器确认它的文件副本是最新的。其次,目前的超高速缓冲存储器信任每一个人,使得攻击(如向超高速缓冲存储器中插入虚假的文件版本)是非常容易实现的。最后,超高速缓冲存储器不向发行商提供反馈,使发行商无法进行用户跟踪。

大多数缓冲存储技术针对的是普通的 Web 对象。最近的一些研究表明专门用于特殊对象的超缓冲存储策略有助于提高总体性能[46]。因此,人们在这一方向上做了大量的研究[46,58,62,91]。缓冲存储技术在视频的一种直接推广,是将全部的视频序列存储在超高速缓冲存储器中。然而,由于大规模的视频数据量和代理服务器上可能有限的超高速缓冲存储空间,这种方法可能是不实用的。作为替代的方法,已经证明即使少数缓冲存储的帧也能够明显地改善性能[46]。Miao 和 Ortega 提出了两种视频缓冲存储策略:初始缓冲存储和选择性缓冲存储,这两种方法在超高速缓冲存储器中存储部分视频流。特别是已经证明,选择性缓冲存储能够最大限度地提高视频流对抗网络拥塞的健壮性,同时又不改变解码器的缓冲区容量。

在这一节,我们介绍了流媒体的三种网络支持机制。下面,我们讨论流服务器设计中的关键问题。

15.5 流服务器

流服务器在提供流服务中起着关键的作用。为了提供高质量的流服务,要求流服务器在定时限制下处理多媒体数据,以便在客户播放期间不会出现人为失真(例如视频运动的抖动和音频的爆裂声)。此外,流服务器必须支持类似 VCR 的控制功能,如停止、暂停/重新开始、快进和快退等。而且,流服务器必须以同步方式读取各媒体内容。例如,读取一个演讲图像,需要把视频和音频与演讲幻灯片同步。

一个流服务器通常包含下面三个子系统:

1. **通信程序:** 通信程序涉及到服务器中实施的应用层和传输协议(见图 15.1)。通过通信

程序、客户可以与服务器通信并以连续的、同步的方式读取多媒体内容。我们已在15.3节介绍过应用层，传输协议将在 15.7 节介绍。

2. **操作系统**：与传统的操作系统不同，流服务器的操作系统必须满足流应用的实时需求。
3. **存储系统**：流服务器的存储系统必须支持连续媒体存储和读取。

下面，我们将分别在 15.5.1 节和 15.5.2 节讨论流媒体的操作系统支持和存储系统。

15.5.1 实时操作系统

操作系统提供与基本资源，如 CPU、内存、存储器和所有输入输出设备有关的各种服务。在下面几节中，我们讨论实时操作系统特有的问题，并综述解决流服务所引入的问题的有关方法。我们首先说明过程管理如何考虑流媒体提出的定时要求以及如何应用恰当的调度方法；然后描述如何管理资源以满足定时要求；最后我们讨论文件管理问题。

进程管理 过程管理安排主处理器资源。过程管理器根据特定的调度策略把单个过程映射到 CPU 资源，使所有过程能够满足的它们的要求。

为了实现连续媒体的定时要求，操作系统必须使用实时调度技术。大多数解决实时调度问题的努力只是多媒体系统的两种基本算法的变种：最早截止时间优先（EDF）[42]和单调请求率调度[14]。在 EDF 调度中，对每一个任务都指定一个截止时间，按截止时间的增长顺序依次对各个任务进行处理。在单调请求率调度中，每一个任务根据它的请求率确定一个固定的优先级①。特别地，周期最短（或请求率最高）的任务的优先级最高，周期最长（或请求率最低）的任务的优先级最低，然后按照优先级的顺序对各个任务进行处理。

EDF 和单调请求率调度都具有优先权。也就是说，调度器可以抢先正在运行的任务，并且根据任务的截止时间或优先级为处理器安排新的任务。中断的任务稍后再重新开始执行。EDF 与单调请求率调度的区别如下所述。EDF 调度器是基于单优先级任务队列，处理器运行具有最早截止时间的任务。而单调请求率调度器是一个具有多优先级任务队列的静态优先权调度器，也就是说，直到所有在较高优先级队列中的任务服务完毕后，才执行在较低优先级队列中的任务。在图 15.7 的例子中，有两个任务序列：高请求率任务序列由任务 1～8 组成；低请求率序列由任务 A～D 组成。如图所示，在单调请求率调度中，任务 2 抢先任务 A，因为任务 2 具有较高的优先级；然而，在 EDF 中，任务 2 不抢先任务 A，因为任务 2 与任务 A 具有相同的截止时间（$dA = d2$）。可见，单调请求率调度器比 EDF 更倾向于任务切换。总的来看，若处理器的利用率低于 69%，那么单调请求率算法能保证满足所有的截止时间[14]；EDF 算法能够达到 100% 的处理器利用率，但不能保证处理某些过载期间的任务。

资源管理 多媒体服务器的资源包括 CPU、内存、存储设备，等等。由于资源是有限的，多媒体服务器只能以所要求的 QoS 服务有限数目的客户。因此，需要资源管理来处理资源以满足定时要求。资源管理包括接纳控制和资源分配。具体地说，在接纳一个新客户前，多媒体服务器必须进行接纳控制测试，以确定新的连接是否可以被接纳，而不会影响为现有的连接所提供的性能保证。如果一个连接被接受，资源管理器就将为这个新的连接分配满足 QoS 所需的资源。

接纳控制算法可分为两类：确定的[24]和统计的[76]。确定的接纳控制算法向客户提供

① 我们假设每一项任务都是周期性的。

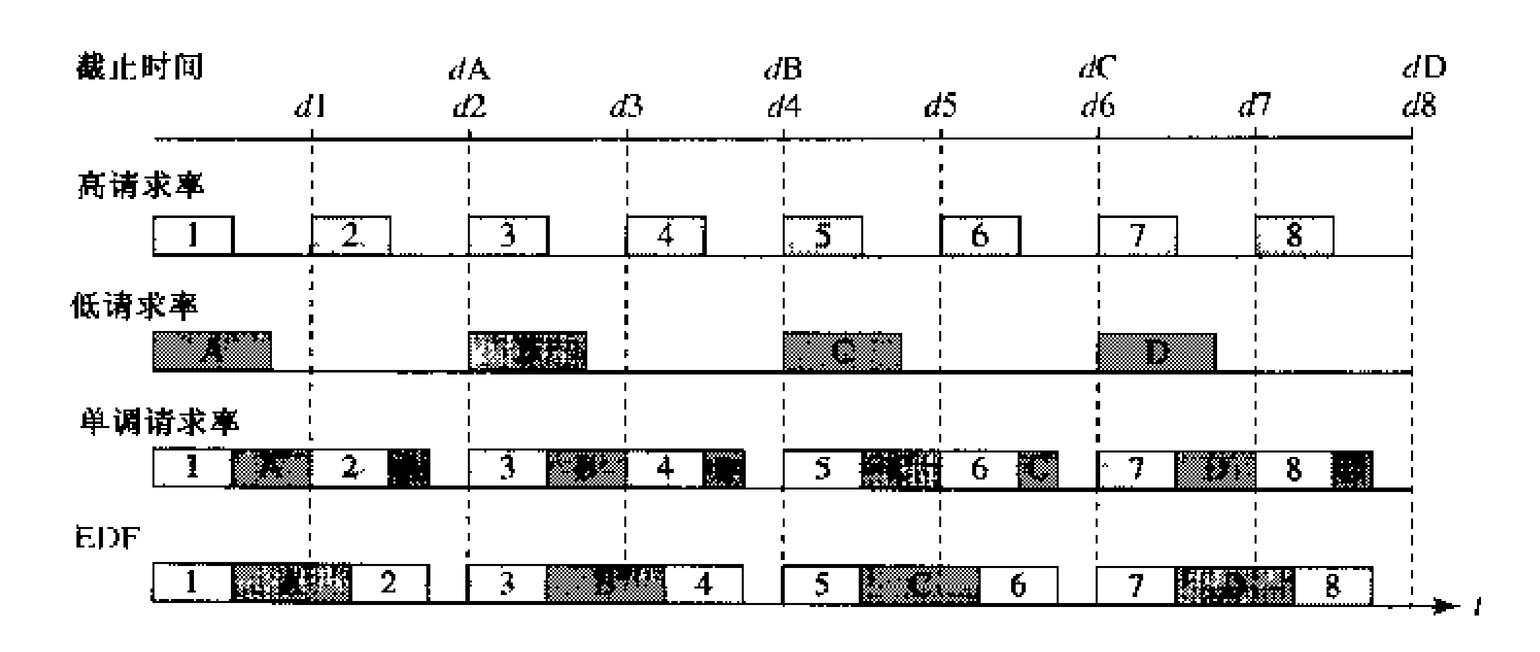


图 15.7　EDF 与单调请求率调度器的对比

硬性保证；而统计算法提供统计的保证（即至少满足某一固定百分比的媒体单元的连续性需求）。确定的接纳控制的优点是简单性和严格的质量保证；其局限性是较低的服务器资源利用率。相反，统计的接纳控制提高了服务器资源利用率，它利用了人的感觉容限以及多媒体服务器在平均和最坏情况下的性能差异[76]。

相应于接纳控制算法，资源分配方案也可以是确定的和统计的。确定的接纳控制算法需要确定的资源分配方案；而统计的接纳控制算法需要统计的资源分配方案。确定的资源分配方案为最坏的情况留有余地。例如，为也许会出现的任务所需要的最长处理时间和最高速率预留带宽。然而，统计的资源分配方案允许暂时的过载，从而达到较高的资源利用率，但这会导致一个小百分比的 QoS 破坏。

文件管理　文件系统为文件的存储和读取提供了访问和控制功能。有两种支持文件系统中连续媒体的基本方法。在第一种方法中，磁盘文件的组织保持它作为离散数据的形式（即一个文件不分散在几个磁盘上），利用由特殊的磁盘调度算法所提供的必要的实时支持和足够的缓存能力，来避免图像的抖动。第二种方法是把视频和音频文件组织在分布式的存储器上，如磁盘阵列。在第二种方法下，通过把每个音频/视频文件分散或条形化存放在多个磁盘上可以提高磁盘的吞吐量，利用磁盘调度算法可以减小磁盘的搜索次数。

传统的磁盘调度算法，如先到先服务和 SCAN[16,70]，并不能提供实时保证。因此，提出了许多支持连续媒体的磁盘调度算法。它们包括：SCAN-EDF[56]、分组扫描调度（GSS）[88]和动态循环 SCAN（DC-SCAN）[31]，分别介绍如下。

- SCAN-EDF 结合了传统的磁盘调度方法 SCAN 的搜索最优化和 EDF 机制的实时保证[16]。注意，磁盘调度中的 EDF 机制是没有优先权的，不同于用于进程管理中的有优先权的 EDF 方案。
- GSS 将集合中的 n 个流分成 g 个组，分组的方法是，具有相似的截止时间的所有流属于同一组；一个组内的各个流按照 SCAN 进行服务。
- DC-SCAN 采用循环的 SCAN[64]服务顺序，使磁盘搜索的开销和交互服务时间的波动最小化，从而得到高的吞吐量。通过动态地调节循环 SCAN 的服务顺序减少启动延迟。

结果是，SCAN-EDF，GSS 和 DC-SCAN 三种算法能够提高连续媒体数据吞吐量，并且满足连续媒体所提出的实时要求。

文件管理所必须支持的另一个功能是交互式控制,如暂停/重新开始、快进和快退。暂停/重新开始操作对高效缓存管理方案的设计形成了重大的挑战,因为它们妨碍了不同的观众对多媒体流的共享,这个问题还有待于进一步研究。快进和快退操作可以通过以高于正常速度播放媒体实现,也可以通过在以正常速度播放的同时跳过某些数据来实现。由于前一种方法会显著增加数据率,因而它的直接实现是不实用的。另一方面,如果存在数据之间的依赖性(例如在 MPEG 中,P 帧和 B 帧取决于 I 帧),那么后一种方法必须小心地设计[12]。对于流 MPEG 视频,在快进操作时必须跳过整个图像组(GOP),观众看到的是普通清晰度的带有间隙的视频,但这是可以接受的。

15.5.2 存储系统

多媒体存储系统设计的挑战性问题是高吞吐量、大容量和容错性。

数据条形化:一个提高吞吐量的方法 如果一个完整的视频文件存放在一个磁盘上,那么对这个文件同时访问的数目就会受到磁盘吞吐量的限制。这就规定了能够观看同一个视频文件的客户数。为了克服这个限制,数据条形化的概念被提出来[63]。在数据条形化方案下,一个多媒体文件分散在多个磁盘上,并且可以并行访问这个磁盘阵列。一个数据条形化的例子示于图 15.8。如图 15.8 所示,文件 A 的块 1、块 2 和块 3 可以并行读取,从而加大了吞吐量。数据条形化方案设计中一个重要问题是平衡最重负载磁盘的负荷以避免过载的情况,同时保持低的等待时间。设计者必须在负荷平衡与等待时间之间进行折中,因为它们是两个冲突的目标[63]。注意,数据条形化不同于文件复制(一种昂贵的提高吞吐量的方法),数据条形化技术只允许在磁盘上存储一个视频文件的副本,而文件复制允许在磁盘上存储一个视频文件的多个副本。

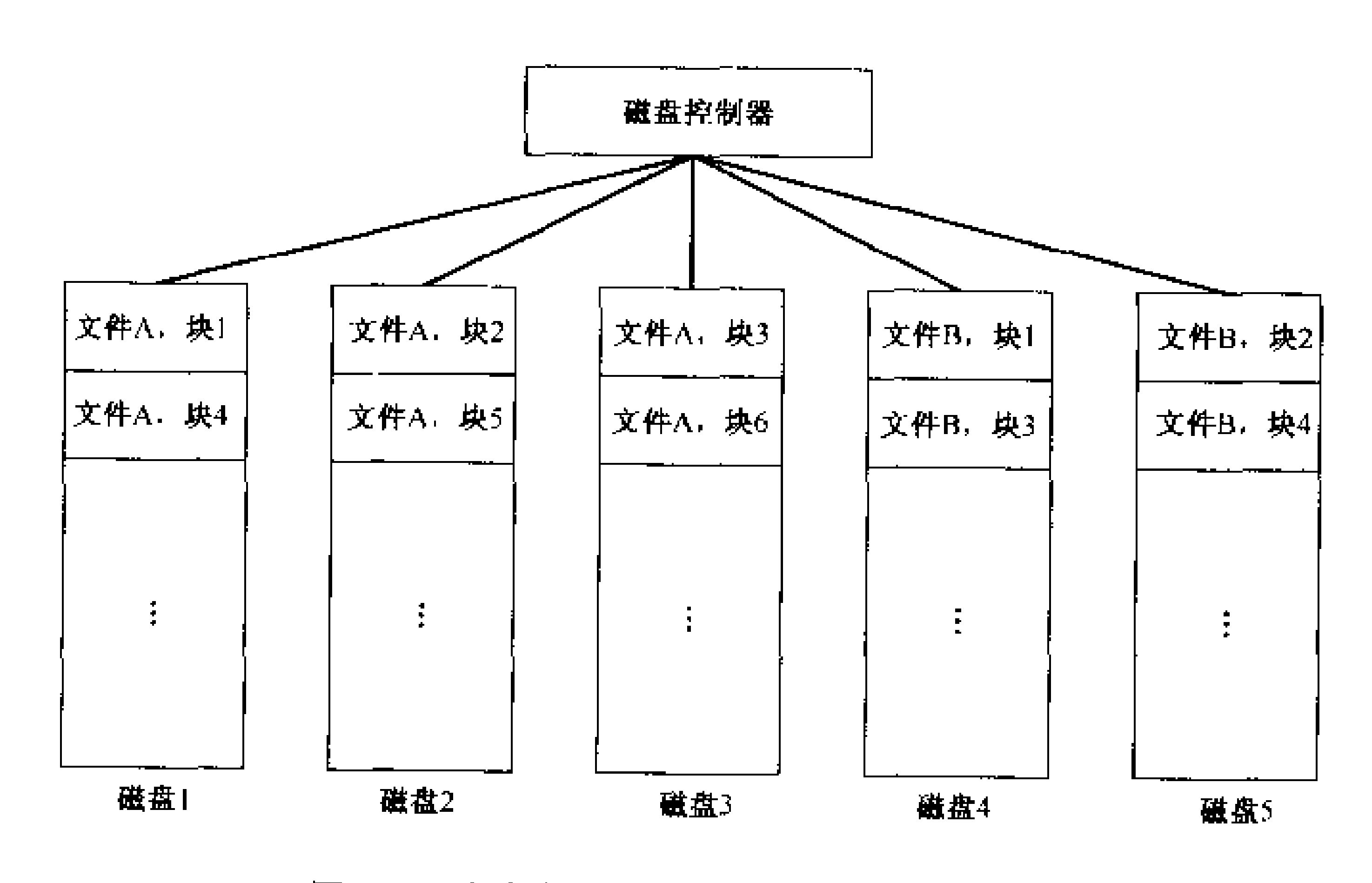


图 15.8 在多个磁盘上的数据条形化和并行访问

第三级和分层存储:一个提高容量的方法　引入多个磁盘可以提高存储容量,如图 15.9 所示。然而,如果大量的磁盘用于存储,那么对于大的档案文件(例如具有 40 万亿字节的存储要求)价格是过高的。为了保持较低的存储成本,需要加入第三级存储(例如自动磁带库或 CD-ROM 光盘机)。

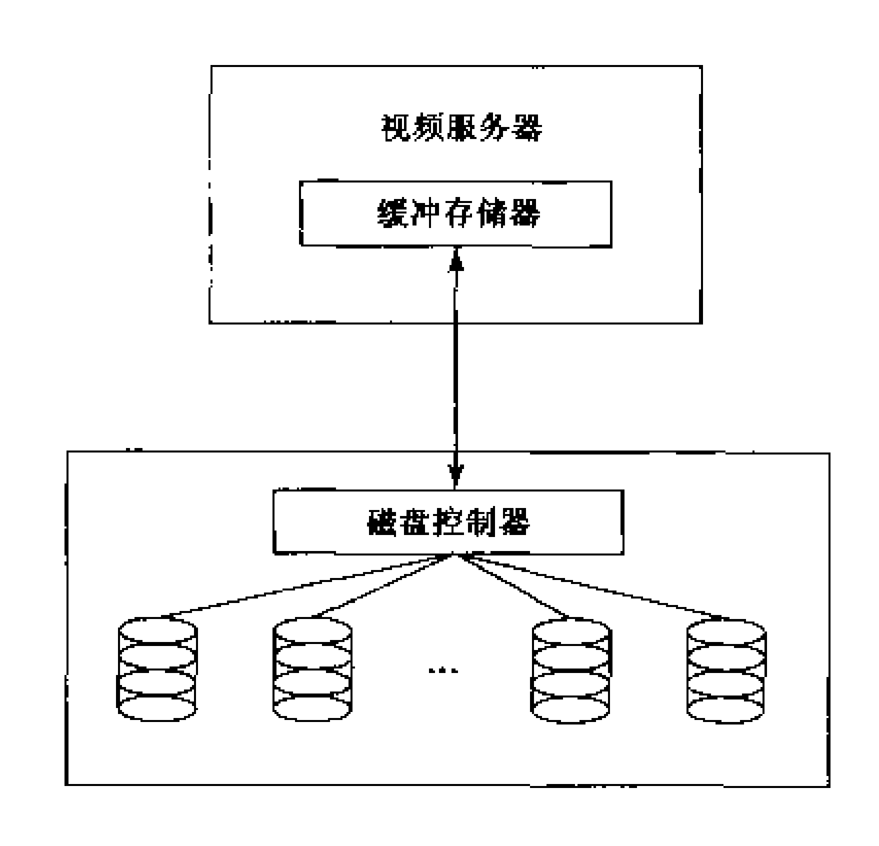


图 15.9　基于磁盘的视频存储

为降低整体成本,通常采用分层存储结构,如图 15.10 所示。在分层存储结构下,只有总存储量的一小部分保存在磁盘上,而其余的主要部分存储在第三级磁带系统。具体地说,频繁请求的视频文件存储在磁盘上以便快速访问,其余的存储在自动磁带库上。

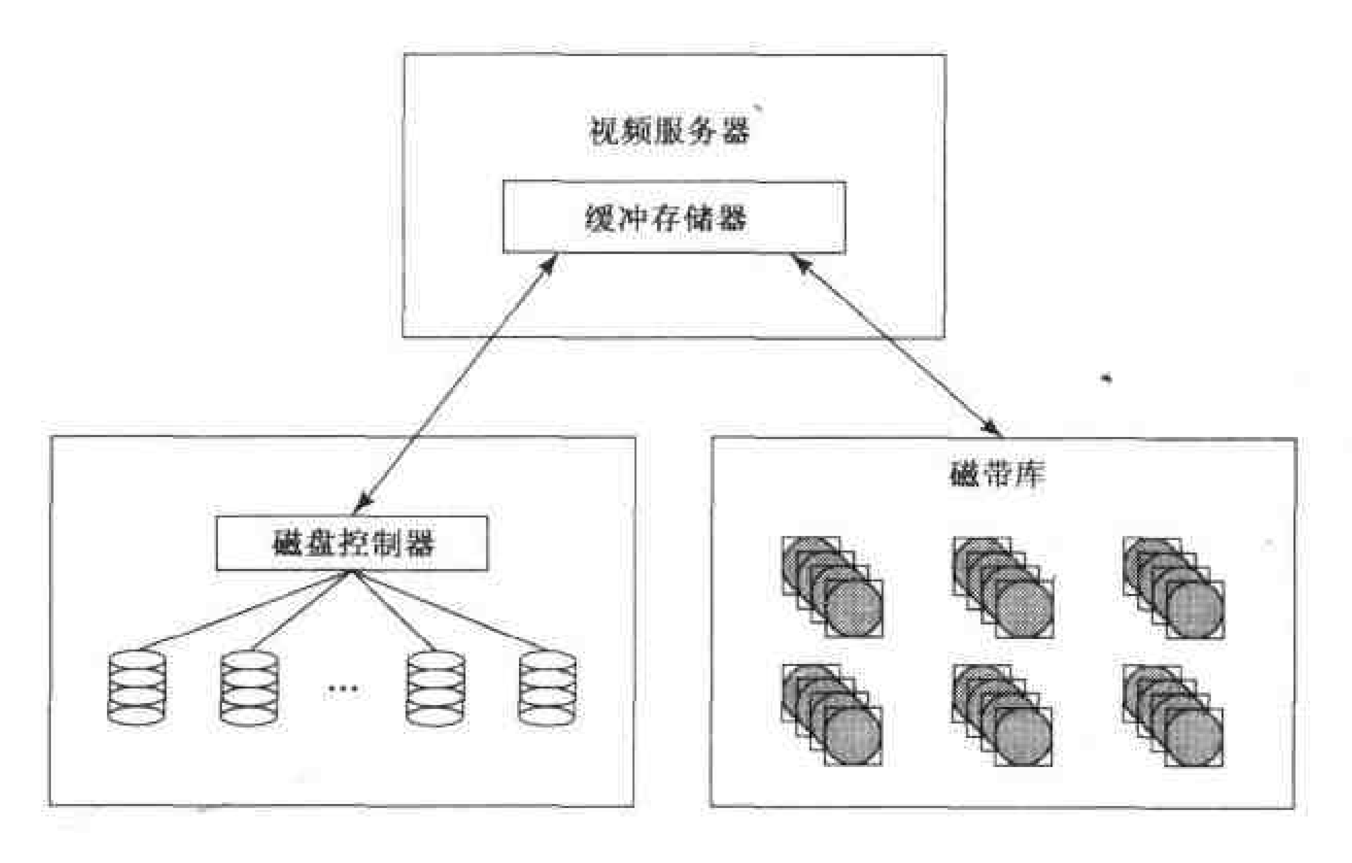


图 15.10　分层存储

为了开展大规模的流服务,必须使用存储区域网 (SAN)结构,如图 15.11 所示[17,29]。SAN 能在存储设备与主机之间提供高速数据管道,该主机比传统主机配属的 SCSI(小型计算机系统接口)具有远得多的距离。在 SAN 中的连接,可以是通过光纤通道仲裁环路 (FC-AL)在专用存储设备和各个主机之间建立的直接链接;或者它们可以通过光纤通道交换技术形成一个矩阵。通过这些高速的连接,SAN 能够在异类的存储设备(例如磁盘阵列、磁带库和光盘存

储阵列)以及多个服务器和存储器客户之间提供多对多的联系。

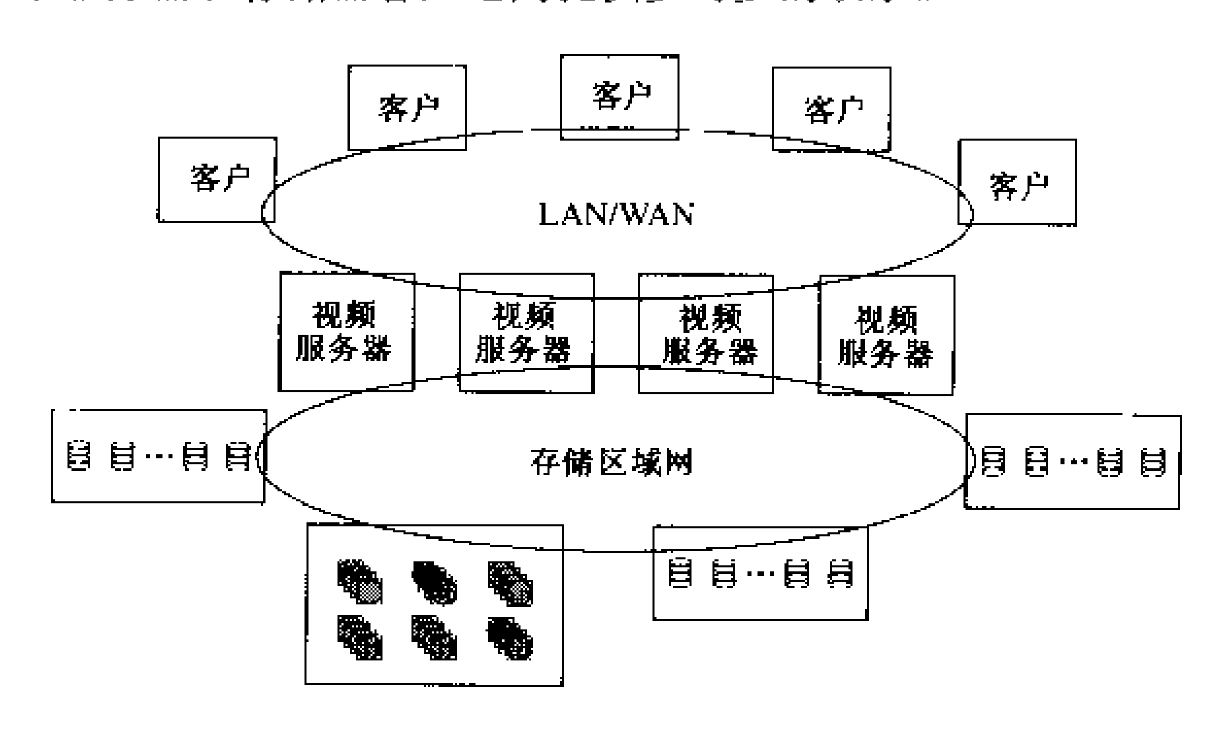


图 15.11 一种基于 SAN 的大规模配置的服务器和存储器结构

容错性 为了保证在磁盘出现错误的情况下不中断服务,服务器必须能重建丢失的信息。这可用冗余信息实现。冗余信息可以是由类似 FEC 的纠错码生成的奇偶校验数据,也可以是存储在分离的磁盘上的备份数据。有两种容错技术:纠错编码(即奇偶校验编码)[4,53,72]和镜像存储[48]。奇偶校验数据增加很少的存储器开销,但它们需要同步读取和附加的处理时间以解码丢失的信息。相反,镜像存储不需要同步读取和附加的处理时间以解码丢失的信息,这样就大大简化了视频服务器的设计和实现。然而,与无容错的情形相比,镜像存储至少造成两倍的存储空间。结果是,需要在可靠性和复杂性(成本)之间做出折中。最近的研究[23]表明,对于相同程度的可靠性,从每一个数据流的成本和磁盘错误后重新开始的等待时间来看,基于镜像存储的方案总是胜过基于奇偶校验的方案。

总之,我们已经介绍了流服务器设计中的各种问题,并给出了高效的、可分级的和可靠的存储和读取多媒体文件的重要技术。下面,我们讨论流媒体的同步机制。

15.6 媒体同步

多媒体应用与传统的数据应用相区别的一个主要特征是必须以同步的方式演播各种媒体流的集成。例如,在远程教育中,幻灯片的显示应该与解说音频流同步(见图 15.12)。否则,当前显示在屏幕上的幻灯片就会与学生们听到的讲解相脱节。通过媒体同步,接收端能够以媒体原始采集的形式演播它们。

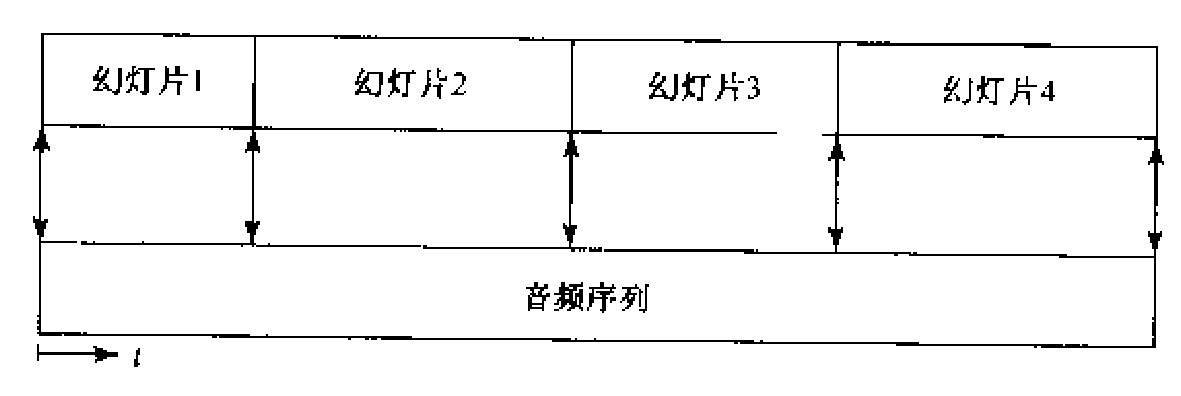


图 15.12 幻灯片与解说音频流之间的同步

媒体同步指的是保持在一个数据流内和在各种媒体流之间的时间关系。有三级同步，分别是流内、流间和对象间同步。这三级对应于如下多媒体数据的三个语义层[66]：

1. 流内同步：连续媒体或与时间有关的数据（如音频和视频）的最低层是媒体层。媒体层的单元是逻辑数据单元（LDU），如视频/音频帧，它们都有严格的时间限制以保证播放时可接受的用户接收效果。这一层的同步是指流内同步，它维持 LDU 的连贯性。如果没有流内同步，数据流的演播就会被暂停或间隙所打断。
2. 流间同步：与时间有关的数据的第二层是流层。流层的单元是整个流。这一层的同步是指流间同步，它维持不同连续媒体之间的时间关系。如果没有流间同步，流间的偏移会变成不可容忍的，例如，如果用户发现说话者嘴唇的运动与播出的音频不一致，他们会感到很不方便。
3. 对象间同步：多媒体文件的最高层是对象层，它集成了流和与时间无关的数据，如文本和静止图像。这一层的同步指的是对象间同步。对象间同步的目的是，若与时间有关的多媒体对象到达了某些预定的演播点，则在一个可允许的时间间隔内开始或停止与时间无关数据的显示。例如播放幻灯片时，若没有对象间同步，如果声音是一个幻灯片的而同时播放的是另一个幻灯片，那么观众就会感到很烦恼。

媒体流在从服务器传输到客户后可能失去同步。如图 15.1 所示，在把数据从存储器传输到用户的路径上有许多环节。具体地说，服务器从存储设备读取数据并把数据发送到网络；网络把数据传输到客户端；客户端从自己的网络接口读出数据并把它送给用户；操作系统和协议允许这些系统运行和工作。传输路径上的每一个这样的环节执行一个特定的任务，并且以不同的形式对数据造成影响。它们都必然以可预测或不可预测的方式引入延迟或延迟变化（抖动）。特别地，由网络引入的延迟一般是不可预测的，这是由因特网最尽力的特性所决定的。产生的延迟或延迟变化会破坏流内、流间和对象间的同步。因此，为了保证在客户端正确地提供多媒体演播，需要媒体同步机制。

任何媒体同步机制的实质部分是对媒体内部和媒体之间时间关系的规定。这种时间关系可以自动或手工设定。在音频和视频录制及回放场合，时间关系是由录制设备自动设定的。在演播是由独立采集的或其他方法生成的媒体组成的场合，时间关系必须手工设定（用人工干预）。可以用幻灯片的设计来说明手工设定的过程：设计者选取合适的幻灯片，创建音频对象，并且定义幻灯必须播放的音频流的段（见图 15.12）。

用于设定时间关系的方法包括基于时间间隔、基于轴、基于控制流和基于事件的设定[7]。一个连续媒体广泛采用的方法是基于轴的设定，或时间戳：在信源端，一个流被打上时间戳，以便在流的内部及相对于其他的流建立时间信息关系；在信宿端，应用系统根据流的时间关系进行演播。

除了设定时间关系外，还希望传输路径上的每个环节都支持同步。例如，服务器对大量数据的存储方式必须能快速和高效地读出，以减小延迟；网络要提供足够的带宽，并且网络所引人的延迟和抖动对于多媒体应用是可容忍的；操作系统和应用系统必须支持实时数据处理（如读取、再同步和显示）。然而，在目前的因特网上，还不具有网络的实时支持，因此大部分同步机制要基于终端系统。这些同步机制可以是预防机制或校正机制[36]。

● 预防机制的作用是在数据从服务器到用户传输时使同步差错最小化。换句话说，预防

机制试图使等待时间和抖动最小化。这些机制包括磁盘读取调度算法、网络传输协议、操作系统和同步调度器。磁盘读取调度是对从存储设备读取的数据进行组织和整理的过程。网络传输协议为数据在因特网上传输期间提供保持同步的方法。操作系统通过EDF或单调率调度实现定时约束的精确控制。同步调度器可以对一个演播使用同步设定，创建一个服务器把媒体流传送到客户的时间表(传送时间表)以及客户应用系统把这些媒体流向用户演播的时间表(演播时间表)。调度器可以是集中的(全部位于客户端)，也可以是分布式的(传送调度功能由服务器和客户分担)。

- 校正机制的设计是在出现同步差错时恢复同步。由于因特网引入的随机延迟，同步差错是不可避免的。随机延迟破坏媒体流的连续性，在数据传输期间造成间断和抖动。因此，当发生同步差错时，在接收端需要一定的补偿措施(即校正机制)。

  校正机制的一个例子是流同步协议 (SSP)[25]。在 SSP 中，“有意延迟”的概念被各种流所使用，目的是调节它们的演播时间，抵消网络延迟的变化。SSP 的工作过程如下所述。在客户端，控制和监视数据连接的客户端单元将数据的实际到达时间与演播时间表上的预报时间进行比较，并把任何差异通知给调度器，调度器对这些差异进行补偿，对比其他数据“提前”到达的数据进行延迟显示，使落后的数据“追上”来。

总之，媒体同步是媒体流服务设计中的一个关键问题。在同步领域人们已经进行了大量的研究。作为这一领域的概述，我们介绍了同步的概念、要求和方法。要获得媒体同步方面更多的信息，请查阅参考文献[7,66]。

15.7 流视频协议

协议是为了在客户机和流服务器之间进行通信而设计和标准化的。根据它们的功能，与因特网上的流视频直接相关的协议可以分为三类：

1. **网络层协议**：网络层协议提供了基本的网络服务支持，如网络寻址。因特网协议(IP)就是用于因特网视频流的网络层协议。
2. **传输协议**：传输协议为流应用系统提供端对端的网络传输功能。这些协议包括 UDP，TCP，RTP 和 RTCP。UDP 和 TCP 是低层传输协议，而 RTP 和 RTCP[59]是高层传输协议，它们是在 UDP/TCP 之上实现的(见图 15.13)。
3. **话路控制协议**：话路控制协议定义消息和程序，以控制多媒体数据在已建立的通话期间传送。RTSP[61]就是这样一种话路控制协议。

为了说明这三种类型的协议之间的关系，我们在图 15.13 中画出了媒体流的协议栈。如图所示，在发信方，压缩的视频和音频数据被读出，并在 RTP/RTCP/RTSP 层上打包，以提供定时和同步信息以及包的序列号。然后把这些打包的 RTP 流送到 UDP/TCP 层和 IP 层。得到的 IP 包在因特网上传输。在收信方，媒体流在演播前按相反的方式处理。这是数据面的情况。对于控制面，RTCP 包和 RTSP 包在 UDP/TCP 层上复用，并且被送到 IP 层，以便通过因特网传输。

这一节的其余部分安排如下：在 15.7.1 节讨论流媒体的传输协议；15.7.2 节介绍话路控制协议，即 RTSP。

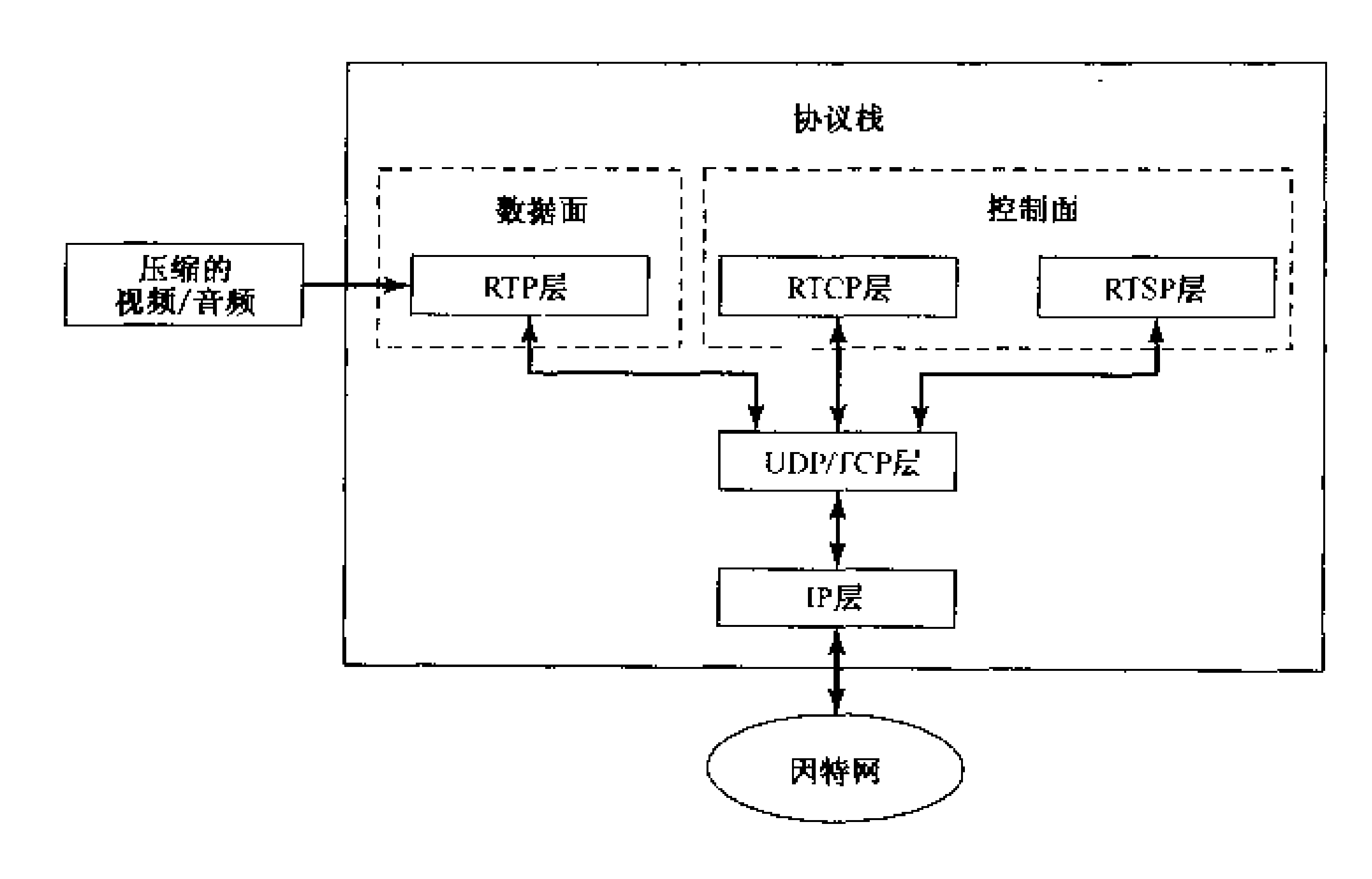


图 15.13　媒体流的协议栈

15.7.1　传输协议

媒体流的传输协议包括 UDP,TCP,RTP 和 RTCP 协议。UDP 和 TCP 提供基本的传输功能，而 RTP 和 RTCP 则运行在 UDP/TCP 之上。

UDP 和 TCP 协议支持的功能包括复用、差错控制和流量控制。这些功能可以简略地描述如下。UDP 和 TCP 能够对来自不同应用程序的数据流进行复用,这些应用程序是运行在具有相同 IP 地址的相同计算机上的。为了差错控制的目的,TCP 和大多数 UDP 实现是利用校验和来检测误码。如果在接收的包中检测到有一个以上的误码,TCP/UDP 层就丢掉这个包,这样上一层(例如 RTP)将不会收到这个损坏的包。另一方面,与 UDP 不同,TCP 用重传来恢复丢弃的包,因此,TCP 提供可靠的传输,而 UDP 则不是。TCP 利用流量控制,根据网络的拥塞状况调节传输码率。这是 TCP 另一个区别于 UDP 的特性。

由于 TCP 重传所引入的延迟对于具有严格延迟要求的流应用来说是不可接受的,因此一般用 UDP 作为视频流传输协议。另外,由于 UDP 不能保证包的传输,所以接收端必须依靠上一层协议(即 RTP)来检测包的丢失。

RTP 是一个因特网标准协议,用于提供端对端的传输功能,以便支持实时应用[59]。RTCP 是 RTP 的同伴协议。设计 RTCP 是为了向 RTP 话路的参与者提供 QoS 反馈。换句话说，RTP 是一个数据传输协议,而 RTCP 是一个控制协议。

RTP 并不保证 QoS 或可靠性传输,而是提供以下支持媒体流的功能:

- **时间戳**:RTP 提供时间标记,用于不同媒体流之间的同步。注意,RTP 本身并不负责同步,同步必须由应用系统实现。
- **序列编号**:由于到达接收端的数据包可能是不按次序的（UDP 不按次序传送数据包），RTP 用序列编号对接收到的数据包进行正确的排序。序列号还可以用于丢包检测。
- **有效载荷类型识别**:包含在 RTP 包中的有效载荷类型由一个称为有效载荷类型识别符的 RTP 包头域来指示。接收端基于有效载荷类型识别符解释包的内容。某些常用的有

效载荷类型如 MPEG-1/2 视频和音频已经指定了有效载荷类型号[60]。对其他的有效载荷，可以由话路控制协议进行指定。

- **信源识别**：每一个 RTP 包的信源由一个称为同步信源识别符(SSRC)的 RTP 包头域来指示。SSRC 为接收端提供了一种区分不同信源的方法。

RTCP 是与 RTP 协同工作的控制协议[59]。在 RTP 话路中，参与者定期发送 RTCP 包，传达关于数据传输质量和成员数信息的反馈。RTCP 提供的服务如下：

- **QoS(QoS)反馈**：这是 RTCP 的首要功能。RTCP 为应用系统提供关于数据发送质量的反馈。这些控制信息对发信方、收信方和第三方监控者是有用的。根据收信方报告的反馈，发信方可以调节发送速率(见 15.3.1 节)。收信方可以确定拥塞是本地的、区域性的还是全局性的。网络管理者可以为组播分配评估网络的性能。

  利用发信方和收信方的报告，RTCP 分别在信源和信宿提供 QoS 反馈。这些报告可以含有关于接收质量的信息，如(1)自从上一次报告后，RTP 丢包的百分率，(2)自从接收开始以来累计的丢包数，(3)包到达时间的抖动，(4)自从接收到发信方最近的报告以来的延迟。

- **参与者识别**：信源可以由 RTP 包头的 SSRC 域来识别。但是 SSRC 识别符对于用户并不方便。为补救这一问题，RTCP 提供了一个人类友好的机制用于信源的识别。具体地说，RTCP SDES(信源描述)包中含有称为规范名的文本信息，作为话路参与者的全局惟一的识别符。规范名包含用户的姓名、电话号码、电子邮箱地址和其他信息。
- **控制包定标**：为了给发送到若干参与者的 RTCP 控制包定标，所设计的控制机制如下。控制机制保持所有的控制包占话路总带宽的 5%。在控制包中，25%分配给发信方报告，75%分配给收信方报告。为了防止控制包缺乏，收信方或发信方至少每隔 5 秒发送一个控制包。
- **媒体间同步**：RTCP 发信方报告含有实时指示和相应的 RTP 时间戳。可用于媒体间的同步，如视频中的嘴唇同步。
- **最少话路控制信息**：这项可选功能用于传送话路信息，如参与者的名字。

15.7.2 话路控制协议：RTSP

RTSP 是因特网上流媒体的话路控制协议[61]。RTSP 的一个主要功能是支持类似 VCR 的控制操作，如停止、暂停/重新开始、快进和快退。此外，RTSP 还可以提供选择传输通道 (例如，UDP、组播 UDP 或 TCP)的方法以及基于 RTP 的传输机制。RTSP 既可用于组播，也可用于单播。

RTSP 的另一个主要功能是建立和控制在媒体服务器和客户机之间的连续的音频/视频媒体流。具体地说，RTSP 提供如下操作：

- **媒体检索**：客户机可以请求一个演播说明，并要求服务器建立一个话路以传送所需的媒体数据。
- **邀请媒体服务器参加会议**：可以邀请媒体服务器参加一个会议，播放媒体或录制图像。
- **向现有话路加入媒体**：服务器和客户机可以互相通知可用于已建立的话路的任何附加媒体。

RTSP 为流音频和视频提供的服务与 HTTP(超文本传输协议)为文本和图形所提供的服务相同。它被设计成与 HTTP 具有相似的语法和操作,从而大多数 HTTP 的扩展机制都可加入到 RTSP。

在 RTSP 中,每一个演示和媒体流都被一个 RTSP URL(通用资源定位器)所识别。全面的演示说明和媒体的性质都定义在一个演播描述文件中,它可能包括编码、语言、RTSP URL、目标地址、端口和其他参数。客户可能通过 HTTP、电子邮件或其他方式获得演播描述文件。

总之,RTSP 用于从媒体服务器启动和直接传送流媒体数据;RTP 是流媒体数据的一个传输协议;RTCP 是一个用于监视 RTP 包传输的协议;UDP 和 TCP 是用于 RTP/RTCP/RTSP 包的低层传输协议;IP 为在因特网上传输 UDP/TCP 包提供了一种方法。这些协议的联合提供了一个完整的因特网上的流服务。

15.8　无线 IP 网络上的流视频

近来,宽带无线网络的出现激起了在无线 IP 网络上进行实时视频通信的极大兴趣。然而,在无线网络上实时传输高质量的视频是一个富有挑战性的任务。这主要是因为如下几个问题:

- **带宽波动**:首先,无线通道的吞吐量会因为多径衰落、同频道干扰和噪声的扰动而降低。其次,无线通道的容量会随着基站与移动主机之间距离的变化而波动。第三,当移动的终端在不同网络间运动时(例如,从无线局域网到无线广域网),通道带宽会发生剧烈的变化(例如从每秒几兆比特到每秒几千比特)。最后,当发生基站交接时,基站可能没有足够的无线电资源去满足新加入的移动主机的需求。因此,带宽波动对于无线网络上实时视频传输是一个严重的问题。
- **高误码率**:与有线连接相比,无线通道一般具有大得多的噪声,而且具有小尺度(多径)和大尺度(阴影)衰落,使得误码率(BER)非常高。误码会对视频演播质量造成破坏性的影响。因此迫切需要视频在无线通道上的健壮传输。
- **异类性**:在组播情况下,接收端可能具有不同的要求和特性,如等待时间、视觉质量、处理能力、功率限制(无线和有线)和带宽限制。接收端的要求和特性的异类性使得设计一个高效的组播机制非常困难。

已经证明,可分级视频能适度地应对带宽的可变性(第 11 章)[2,45]。可分级视频编码方案产生一个部分可解码的压缩比特流。与解码全部比特流相比,压缩比特流的部分解码会产生降质的图像,或较小的图像尺寸,或较小的帧率[18]。与此相对照,不可分级视频对带宽波动更为敏感,因为它不能使视频的表示适应带宽的变化[45]。因此,可分级视频更适合用在无线环境下应付无线通道的波动。而且,可分级的视频表示对于组播情况下的异类性问题是一个很好的解决方案[45]。

近来,应用感知的自适应服务已经被证明能够有效减轻无线网络中资源可用性的波动[2]。可分级视频表示自然地适合不平等差错保护,它可以有效抵抗无线媒体所造成的误码。这就促使我们提出自适应帧结构以支持无线 IP 网络高质量的视频通信。

对于无线视频传输,在文献中已经提出了许多自适应方法和服务,其中包括“自适应预留服务”框架[38]、基于 QoS 界限和收益的自适应服务[44]、针对端对端 QoS 设置的自适应帧结

构[49]、效用公平的自适应服务[6]、软 QoS 控制架框[57]、基于一个自适应 QoS 范例的电信业务模型[33]、自适应 QoS 管理结构[37]、无线 IP 网络上可分级视频的自适应框架[84]。

在这一节，我们介绍一个用于将来允许 QoS 的无线 IP 网络的自适应框架[85]。这种自适应框架包括：(1)可分级视频表示，每一级都有自己特定的 QoS 要求，(2)网络感知应用系统，可感知网络的状况，(3)自适应服务，使网络元素支持可分级视频表示的 QoS 需求。在这种框架下，当无线通道条件变化时，移动终端和网络元素能够对视频流分级，并以可接受的接收质量把分级的流传送到接收端。自适应框架具有如下关键特征：

1. 适度的质量退化：与不可分级视频相比，可分级视频可以使它的视频表示适应带宽的变化，并且网络可以根据感知的视频表示进行丢包。因此，在严峻的通道条件下，接收质量被适度地降质。
2. 高效率：当存在剩余带宽（不包括预留带宽）时，剩余带宽会被有效利用，使接收质量或收益最大化；
3. 公平性：资源可以按效用公平[6]或最大最小公平[44]的方式共享。

本章的其余部分安排如下：15.8.1 节介绍网络感知的应用系统、15.8.2 节介绍无线 IP 网络上传输可分级视频的自适应服务。

15.8.1 网络感知应用系统

网络感知应用系统的使用是由下列因素引起的：(1)当通道状况很差时，误码率会非常高，(2)若可用带宽小于所需带宽，丢包是不可避免的。如果发信方在对网络状况一无所知的情况下试图将图像的每一层都发送出去，那么所有的层都会以相等的概率遭到破坏，这样图像质量就会很差。为了解决这个问题，Wu，Hou 和 Zhang[84]提出使用网络感知的应用系统，它根据网络状况以明智的方式抢先丢弃增强层。

为了进行说明，我们在图 15.14 中给出了一个结构，它包括一个网络感知的移动发送器、一个应用感知的基站和一个接收端。图示的结构对现场的和存储的视频都是适用的。在发信方，首先用分级器对压缩的视频比特流进行滤波，从中选取所要传输的某些视频层。然后使所选择的视频表示通过传输协议。在传输到基站之前，比特流必须由调制解调器进行调制。基站收到视频包后就把它们通过因特网发送到目的地。

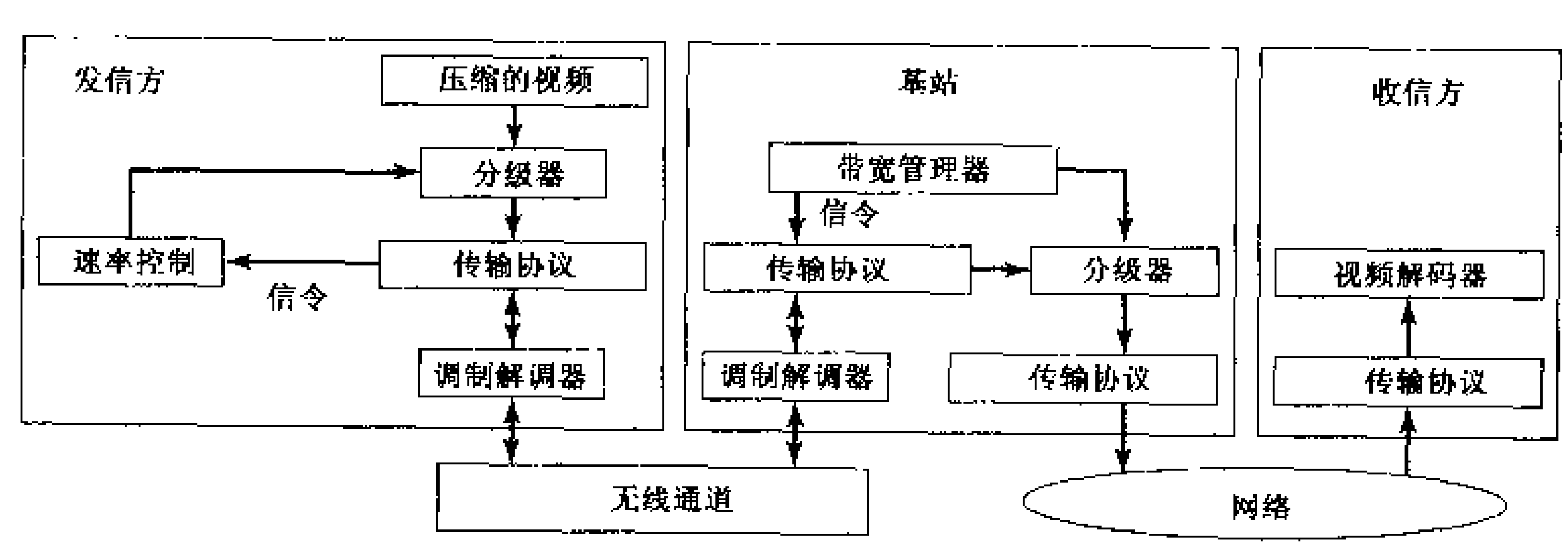


图 15.14 一个从移动终端到有线终端的可分级视频传输结构

注意,分级器能够识别视频层,并根据它们的重要程度丢弃某些层。丢弃的次序是从最高增强层向下到基本层。分级器只执行两种操作:(1)对接收的视频表示向下分级,即丢弃增强层,(2)把接收到的数据都发送出去,即不对接收的视频表示分级。

在这种结构下,基站运用一个带宽管理器。带宽管理器的一个功能是通过一个信令通道来通知发信方关于无线通道的可用带宽[50]。接收到这个信息后,发信方中的速率控制模块把带宽参数传达给分级器。然后,分级器调节视频流的输出码率,使传输速率小于或等于可用带宽。

另一种情况是基站通知发信方关于通道质量(即 BER)的信息[3]。接收到这个信息后,发信方中的速率控制模块命令分级器执行如下操作(假定视频被压缩为两层):如果 BER 高于某个阈值,则丢弃增强层,从而使分配给增强层的带宽可用于前向纠错(FEC);否则就两层都传输。

网络感知的应用系统有两个优点。第一,由于考虑了可用带宽,发信方最好地利用了网络资源,它有选择地丢弃增强层,使得更重要的层被破坏的可能性最小化,从而提高了所传送视频的接收质量。第二,由于考虑了通道的出错状况,发信方可以丢弃增强层,而 FEC 可以利用分配给增强层的带宽来保护基本层,因此使基本层被正确接收的概率最大化。

注意,网络感知的应用系统需要在物理/链接层的自适应技术的支持。这些自适应技术包括联合使用可变扩展、编码和码分多址 (CDMA)系统中的编码集合,自适应编码和时分多址(TDMA)系统中的调制、通道质量估计和一个测量反馈通道[50]。另外,反馈时间间隔一般限制在几十到几百毫秒的数量级[50]。

15.8.2 自适应服务

可分级视频编码器可以为网络生成多个层或多个子流。自适应服务根据固定网或无线网的资源可用性条件提供子流的分级。具体地说,自适应服务包括如下功能:

- 预留最小带宽以满足基本层的需要,因此接收质量总能达到可接受的标准。
- 根据可用带宽和公平原则调节增强层,换句话说,根据资源的可用性对视频流进行分级。

在网络中使用分级的优点包括:

- 对网络异类性的自适应性。例如,当具有较大带宽的上游链接流入具有较小带宽的下游链接时,在连接点处使用分级器有助于改善视频质量。这是因为分级器能有选择地丢弃子流,而不是随机地丢弃包。
- 低延时和低复杂性。可分级视频表示使得分级器的操作非常简单,即只需丢掉增强层。因此比不可分级视频处理要快。
- 较低的呼叫闭塞和基站交接的掉线概率。在基站处可分级视频的自适应性可以转化为较低的呼叫闭塞和基站交接的掉线概率。

自适应服务可应用于整个网络(即端对端设置),或只用于基站(即局部设置)。由于自适应服务的局部设置只是端对端设置的一个子集,所以我们将集中讨论端对端设置。

端对端的自适应服务所需要的部件包括:(1)服务合同,(2)呼叫接纳控制和资源预留,(3)移动组播机制,(4)子流分级,(5)子流调度,(6)链接层差错控制。下面对这些部件进行较详细的描述。

服务合同　一个应用系统和网络间的合同可以包含多个子合同,每个子合同对应于一个或多个具有相似的 QoS 保证的子流。每个子合同必须规定所对应子流的传输特性和 QoS 要求。一个典

型的情况是，用于基本层的子合同规定预留的带宽，而用于增强层的子合同则不规定任何 QoS 保证。

在视频信源处，子流必须根据应用系统所用的子合同生成，并在网络访问点被整形。此外，一个子流按照它的重要性被赋予一个优先级。例如，基本层被赋予最高优先级。优先级用于路由选择、调度、分级和自适应网络的差错控制部件。

呼叫接纳控制和资源预留 在端对端的 QoS 设置中，呼叫接纳控制(CAC)和资源预留是两个主要的部件。CAC 的功能是检测接纳一个进入的连接是否会降低现有连接的服务质量以及是否能满足进入连接的 QoS 要求。如果一个连接请求被接受，就必须为这个连接预留出两部分资源。首先，为了在一个较长的时间尺度上维持指定的 QoS，网络必须沿着移动连接的当前路径预留资源。第二，为了在一个短时间尺度上无缝地实现 QoS，在连接到相邻基站的传输中必须进行一些备份工作，使得在基站发生交接事件时可以避免连接断线。

可分级视频表示(即子流)的概念为 CAC 和资源预留问题提供了一个非常灵活有效的解决方案。首先，因为一般只有基本层子流要求 QoS 保证，所以不必为全部的流预留带宽。这样，CAC 只基于基本层的需求，并只为基本层子流预留资源。其次，多重连接的增强层子流可以共享剩余的带宽。在带宽不足和/或严重差错的条件下，增强层子流是有待于分级的，这将在后续段落中讨论。

移动组播机制 CAC 和资源预留能够提供连接层的 QoS 保证。为了保证在包级无缝的 QoS，需要使用移动组播机制。也就是说，基本层流在沿着它的当前路径传输的同时，还要组播到相邻的基站，以便 QoS 在小的时间尺度上能够无缝地实现。

为了支持无缝的 QoS，移动路由协议必须是主动性的和预见性的，以匹配子流的延迟、丢失和抖动的限制。根据子流的要求，可能需要建立组播路径。组播路径以基站为终点，它们是潜在的移动终端访问候选点。组播路径的覆盖范围取决于移动接收端的 QoS 需求和移动性以及基站交接特性。当一个移动站从一个基站交接到另一个基站时，新的路径被加入，旧的路径被删除[49]。

子流分级 分级用于带宽波动期间和差的通道条件下。当路径上的可用带宽因移动性或衰落而减小时，低优先级的子流被路径上的分级器所丢弃，而高优先级的子流被传输。当较多的带宽变为可用时，较低优先级的子流就可以通过分级器，从而使接收端的接收质量提高。图 15.14 示出了一个从移动终端向有线终端传送可分级视频的结构。图 15.15 画出了一个从有线终端向移动终端传送可分级视频的结构。我们没有示出从一个移动终端向另一个移动终端传送可分级视频的结构，因为它只是图 15.14 和图 15.15 的联合。

分级决定是由带宽管理器做出的。当没有多余带宽(不包括预留带宽)时，带宽管理器命令分级器丢掉增强层。当有多余带宽时，多余带宽可按效用公平[6]或最大最小公平方式[44]共享。

子流调度 子流调度器用于移动终端和基站。它的功能是根据包的子流 QoS 技术要求和优先级在无线媒体上调度要传输的数据包。

当观测到一个短的衰落期时，移动终端试图区分子流传输的优先次序，从而实现最佳的 QoS。根据通道条件，一个子流可能会丢弃一段时间，以便照顾较高优先级的子流。为了确定一个特定的子流中任一包的传输时间(或它在传输队列中的位置)，调度器主要考虑两个因素：该子流与其他子流比较的相对重要性和无线通道条件。

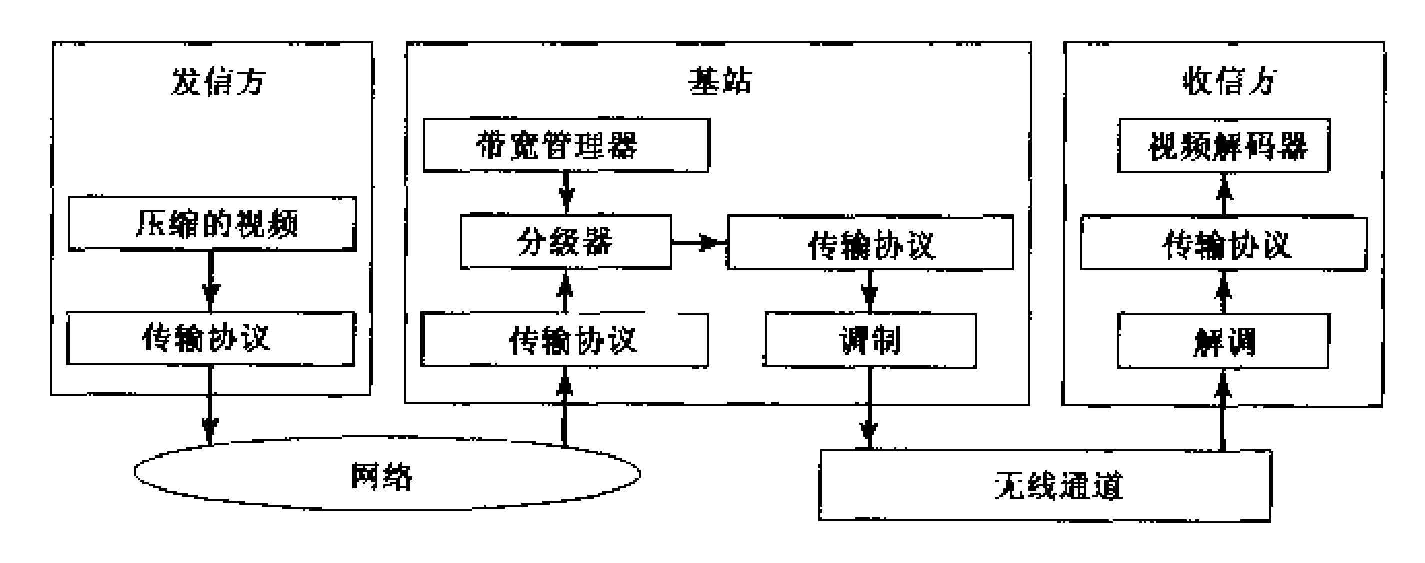


图 15.15 一个从有线终端到移动终端的可分级视频传输结构

为同时实现 QoS(例如有限延迟和预留带宽)和公平性,必须利用包公平排队一类的算法[5]。虽然现有的包公平排队算法能够在有线网络中提供有限延迟和公平性,但它们不能直接用于无线网络。关键的困难是在无线网络中,通话可能经历与位置有关的通道差错。这会导致一个通话得到的服务比预期少很多,而另一个通话得到的服务比预期更多。这导致通话有效时间之间的差异,使它难于同时提供延迟保证和公平性。

为了应用包公平排队算法,Ng,Stoica 和 Zhang[51]确定了无线环境下包公平排队算法应该具备的一系列特性,称为通道条件独立公平(CIF):(1)对无差错通话的延迟和吞吐量的保证,(2)对有差错通话的长期公平性,(3) 对无差错通话的短期公平性,(4)对所接收的超过服务时间的通话适度地降质。后来,他们给出了一个无线网络的自适应包公平排队算法的理论分析,并利用这个理论分析导出了一个基于起始时间的公平排队[28],称为通道条件独立包公平排队(CIF-Q),它实现了前述的所有特性[51]。

作为一个例子,我们考虑两层的视频。基本层子合同规定了预留带宽,而增强层子合同没有规定任何 QoS 保证,这是一般的情况。一个子流调度结构如图 15.16 所示。

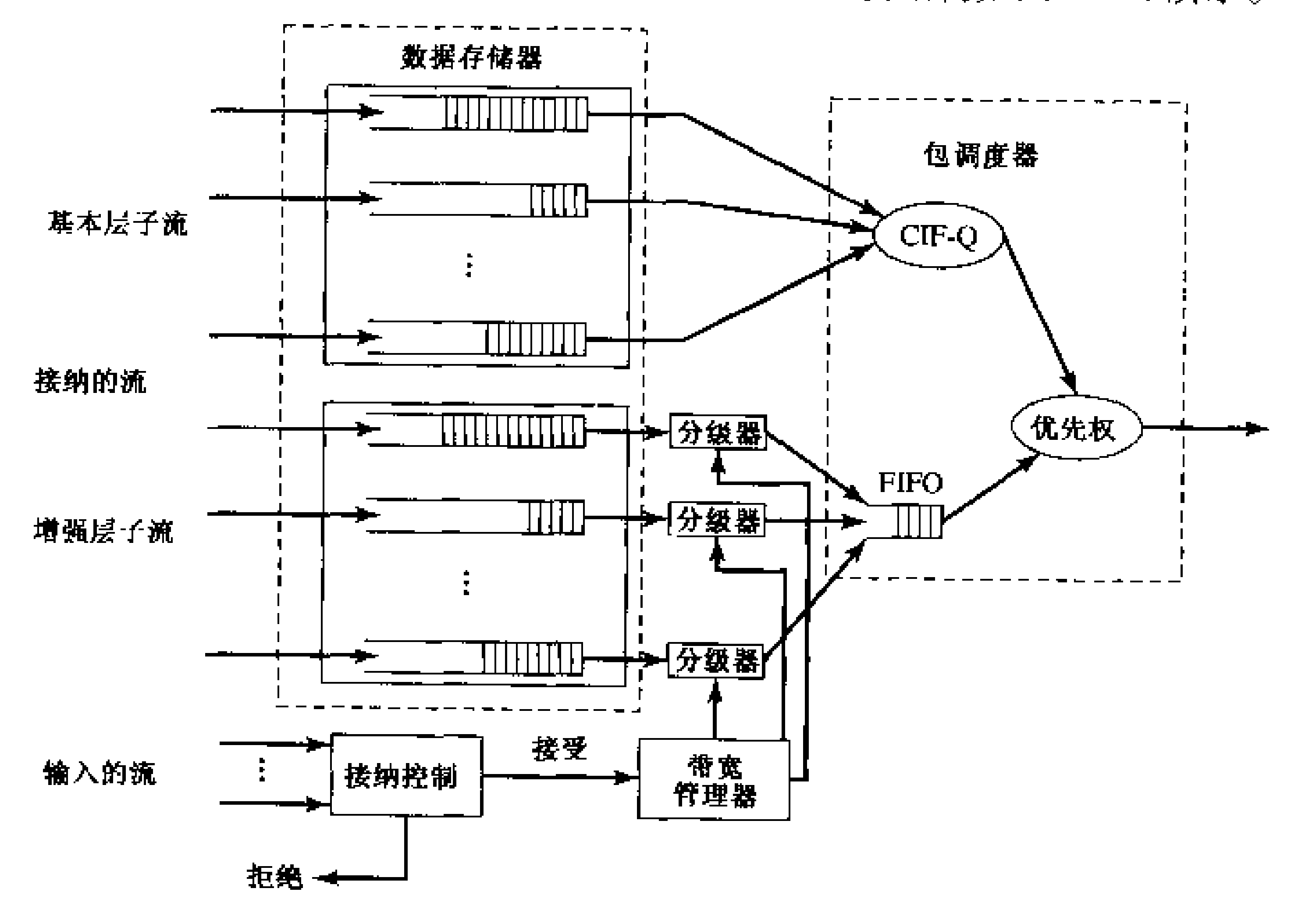


图 15.16 一个基站处的子流调度结构

在这个结构下,缓冲池(即图15.16中的数据存储器)分为两部分:一部分用于基本层子流,一部分用于增强层子流。在同一个用于基本层或增强层的缓冲区分区内,对每一个子流进行总流量排队。而且,在同一个缓冲区分区内的子流共享该分区的缓冲池,而没有跨分区的缓冲区共享。已经证实,这种方法在通信隔离和缓冲区共享之间给出了很好的平衡[84]。

在这种缓冲结构下,Wu,Hou和Zhang[84]设计了基于总流量的通信管理算法,以实现所需要的QoS和公平性。通信管理的第一部分是CAC和带宽分配。视频连接由CAC根据它们的基本层QoS要求接纳,进而为该接纳的基本层子流做出预留带宽。对于接纳的增强层子流,它们的带宽将通过一个带宽管理器动态地分配,这已经在前面的段落中说明了。分级的增强层子流进入共享的缓冲区,并由一个先进先出(FIFO)调度器安排时序。通信管理的第二部分是包调度。图15.16示出了一个包调度的分层结构,其中一个优先权链接调度器被一个用于基本层子流的CIF-Q调度器和一个用于增强层子流的FIFO调度器共享。服务的优先权首先交给CIF-Q调度器,然后再交给FIFO调度器。

链接层差错控制 为了在无线通道上提供优质的视频,需要链接层差错控制。基本上有两种差错控制机制,即前向纠错(FEC)和自动重传请求(ARQ)。FEC的缺点是不能自适应变化的通道条件,而且只当BER低于FEC码的恢复能力时才起作用。ARQ的缺点是延迟的无约束性。也就是说:在最坏的情况下,为恢复误码,一个数据包可能要重发无数次。

为了解决与FEC和ARQ有关的问题,提出了截断II型混合ARQ方案[41,89]和延迟约束混合ARQ[82]。这些混合的ARQ方案结合了FEC和ARQ的优点:延迟约束和自适应性。另一方面,不平等差错保护[30]很自然地适合于可分级视频的分层结构。具体地说,在防止传输差错方面,基本层可以比增强层受到更好的保护。这种不平等差错保护形式比保护所有的子流更为理想。一个开放的问题是如何把不平等差错保护与混合ARQ方案结合起来[85]。

15.9 小结

本章综述了因特网视频流的主要方法和机制,并且介绍了无线IP网络视频的自适应结构。本章的目的不是对现有方法和机制进行详尽考察,而是使读者对一个可用的选择范围和有关的性能、功能和复杂性之间的折中考虑有所了解。

视频压缩(15.2节)

- 可分级视频编码是在因特网和无线IP网络上传输视频的首选模式。根据网络条件,发送端、接收端和中间网络节点决定对视频的哪一层分别进行发送、接收或转发。

应用层服务质量控制(15.3节)

- 这一类技术限制丢包和带宽变化对视频质量的影响。在发送端、网络节点和接收端的拥塞控制,利用码率整形和码率控制机制使视频流适应网络条件。差错控制、错误隐藏、FEC和延迟约束重发使传输差错的视觉效应最小化。

连续媒体发布服务(15.4节)

- 这一节讨论了采用视频内容感知的网络滤波器进行丢帧控制。当用相同的内容服务多个客户时,为有效地利用带宽,可使用组播。应用层组播实现了分离的组播网络的连接。内容复制将内容缓存在网络的几个节点上。然后把内容从最近的高速缓冲存储器传送给用户。

流服务器(15.5 节)

- 流服务器通过网络把内容传送给客户。服务器在定时的约束下处理媒体,以便保证接收方同步不同的媒体,并防止在视频运动中出现抖动之类的人为失真。这需要对服务器的三个主要部分最优化:操作系统、存储系统以及通过一套协议与客户相互作用的通信程序。

媒体同步(15.6 节)

- 媒体同步能够使客户以同步的方式演播音频、视频、图形等不同的媒体流。服务器在所发送的每个包上加上一个时间戳。接收方利用这些时间戳来估计网络抖动以及合适的缓冲区容量,以保证同步性和连续的媒体播放。

流视频协议(15.7 节)

- 基本的传输协议 TCP 和 UDP 在传送包时分别需要和不需要向发信方发送应答。RTP 支持实时应用系统的端对端传输,所提供的功能有时间戳、包编号和有效载荷识别等。RTCP 是相关的控制协议,主要向服务器提供 QoS 反馈。RTSP 控制一个以上的同步媒体流的通信话路。它提供的功能有停止、暂停/重新开始、快进和快退等。

无线 IP 网络上的流视频(15.8 节)

- 由于高的带宽波动和误码率,在无线 IP 网络上的流视频需要在无线终端发送器和无线网络基站中的网络感知应用系统。该系统根据当前的可用带宽对可分级视频比特流进行整形。

我们要强调的是所涉及的七个领域是流视频结构的基本组块。这样一个结构与信号处理、网络技术和服务器技术等有广泛的联系。因此,对流视频整个结构的透彻理解有助于用于流视频的信号处理技术(例如可分级视频压缩)的发展。此外,信号处理和网络技术的深入知识有助于有效地设计和使用应用层 QoS 控制、连续媒体发布服务、协议和无线 IP 网络上的视频服务。而且,这种结构式的理解对设计高效的、可分级的、可容错的流服务器也是有帮助的。

15.10 文献目录

[1] Albanese, A., J. Blömer, J. Edmonds, M. Luby, and M. Sudan. Priority encoding transmission. *IEEE Trans. on Information Theory (Special Issue on Coding Theory)* (Nov. 1996), 42(6, I):1737–44.

[2] Balachandran, A., A. T. Campbell, and M. E. Kounavis. Active filters: delivering scalable media to mobile devices. *Seventh International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'97)*, 125–35, St. Louis, MO, May 1997.

[3] Balachandran, K., S. R. Kadaba, and S. Nanda. Channel quality estimation and rate adaptation for cellular mobile radio. *IEEE J. Select. Areas Commun.* (July 1999), 17(7):1244–56.

[4] Berson, S., L. Golubchik, and R. R. Muntz. Fault tolerant design of multimedia servers. *ACM SIGMOD'95*, 364–75, May 1995.

[5] Bharghavan, V., S. Lu, and T. Nandagopal. Fair queuing in wireless networks: issues and approaches. *IEEE Personal Commun. Mag.* (Feb. 1999), 44–53.

[6] Bianchi, G., A. T. Campbell, and R. Liao. On utility-fair adaptive services in wireless networks. *Sixth International Workshop on Quality of Service (IWQoS'98)*, Napa Valley, CA, May 1998, 256–67.

[7] Blakowski, G., and R. Steinmetz. A media synchronization survey: reference model, specification, and case studies. *IEEE Journal on Selected Areas in Communications* (Jan. 1996), 14(1):5–35.

[8] Bolot, J.-C., T. Turletti, and I. Wakeman. Scalable feedback control for multicast video distribution in the Internet. *ACM SIGCOMM'94*, 58–67, London, Sept. 1994.

[9] Chang, Y.-C., D. G. Messerschmitt, T. Carney, and S. A. Klein. Delay cognizant video coding: architecture, applications, and quality evaluations. Forthcoming in *IEEE Trans. Image Processing*.

[10] Chang, Y.-C., and D. G. Messerschmitt. Adaptive layered video coding for multitime scale bandwidth fluctuations. Submitted to *IEEE Journal on Selected Areas in Communications*.

[11] Chankhunthod, A., P. B. Danzig, C. Neerdaels, M. F. Schwartz, and K. J. Worrell. A hierarchical Internet object cache. *USENIX 1996 Annual Technical Conference*, Jan. 1996, 153–63.

[12] Chen, M., D. D. Kandlur, and P. S. Yu. Support for fully interactive playout in a disk-array-based video server. *ACM Multimedia'94*, San Francisco, CA, Oct. 1994, 391–98.

[13] Cheung, S. Y., M. Ammar, and X. Li. On the use of destination set grouping to improve fairness in multicast video distribution. *IEEE INFOCOM'96*, 2:553–60, March 1996.

[14] Chung, J. Y., J. W. S. Liu, and K. J. Lin. Scheduling periodic jobs that allows imprecise results. *IEEE Transactions on Computers* (Sept. 1990), 19(9): 1156–73.

[15] Davis, G., and J. Danskin. Joint source and channel coding for image transmission over lossy packet networks. *SPIE Conference on Applications of Digital Image Processing XIX*, SPIE-2847:376–87, Denver, Aug. 1996.

[16] Denning, P. J. Effects of scheduling on file memory operations. *AFIPS Spring Joint Computer Conference*, 9–21, Reston, VA, 1967.

[17] Du, D. H. C., and Y.-J. Lee. Scalable server and storage architectures for video streaming. *IEEE International Conference on Multimedia Computing and Systems* (June 1999), 1:62–67.

[18] Ebrahimi, T., and M. Kunt. Visual data compression for multimedia applications. *Proceedings of the IEEE* (June 1998), 86(6):1109–25.

[19] Eleftheriadis, A., and D. Anastassiou. Meeting arbitrary QoS constraints using dynamic rate shaping of coded digital video. *Fifth International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'95)*, 95–106, April 1995.

[20] Fan, L., P. Cao, J. Almeida, and A. Z. Broder. Summary cache: a scalable widearea web cache sharing protocol. *IEEE/ACM Trans. on Networking* (June 2000), 8(3):281–93.

[21] FastForward Networks. FastForward Networks' broadcast overlay architecture. http://www.ffnet.com/pdfs/boa-whitepaper.pdf.

[22] Floyd, S., and K. Fall. Promoting the use of end-to-end congestion control in the Internet. *IEEE/ACM Trans. on Networking* (Aug. 1999), 7(4):458–72.

[23] Gafsi, J., and E. W. Biersack. Performance and reliability study for distributed video servers: mirroring or parity? *IEEE International Conference on Multimedia Computing and Systems* (June 1999), 2:628–34.

[24] Gemmell, J., and S. Christodoulakis. Principles of delay sensitive multimedia data storage and retrieval. *ACM Trans. on Information Systems* (Jan. 1992), 10(1):51–90.

[25] Georganas, N. D. Synchronization issues in multimedia presentational and conversational applications. *Pacific Workshop on Distributed Multimedia Systems (DMS'96)*, June 1996.

[26] Ghanbari, M. Cell-loss concealment in ATM video codes. *IEEE Trans. Circuits Syst. Video Technol.* (June 1993), 3:238–47.

[27] Girod, B., U. Horn, and B. Belzer. Scalable video coding with multiscale motion compensation and unequal error protection. In Y. Wang et al., eds., *Multimedia Communications and Video Coding*. New York: Plenum Press, 1996, 475–482.

[28] Goyal. P., H. M. Vin, and H. Cheng. Start-time fair queueing: a scheduling algorithm for integrated services packet switching networks. *IEEE ACM Trans. Networking* (Oct. 1997), 5(5):690–704.

[29] Guha, A. The evolution to network storage architectures for multimedia applications. *IEEE International Conference on Multimedia Computing and Systems* (June 1999), 1:68–73.

[30] Hagenauer, J., and T. Stockhammer. Channel coding and transmission aspects for wireless multimedia. *Proceedings of the IEEE* (Oct. 1999), 87(10):1764–77.

[31] Hamidzadeh, B., and J. Tsun-Ping. Dynamic scheduling techniques for interactive hypermedia servers. *IEEE Trans. on Consumer Electronics* (Feb. 1999), 45(1):46–56.

[32] Hemy, M., U. Hengartner, P. Steenkiste, and T. Gross. MPEG system streams in best-effort networks. *Packet Video Workshop*, New York, April 1999.

[33] Iera, A., A. Molinaro, and S. Marano. Wireless broadband applications: The teleservice model and adaptive QoS provisioning. *IEEE Communications Magazine* (Oct. 1999), 71–75.

[34] Illgner, K., and F. Mueller. Spatially scalable video compression employing resolution pyramids. *IEEE J. Select. Areas Commun.* (Dec. 1997), 15(9):1688–703.

[35] ISO/IEC. IS14496-2: Information technology—coding of audio-visual objects, part 2: visual. 1999 (MPEG-4 Video).

[36] Jarmasz, J. P., and N. D. Georganas. Designing a distributed multimedia synchronization scheduler. *IEEE International Conference on Multimedia Computing and Systems* (June 1997), 451–457.

[37] Lataoui, O., T. Rachidi, L. G. Samuel, S. Gruhl, and R.-H., Yan. A QoS management architecture for packet switched 3rd generation mobile systems. *Networld+Interop 2000—Engineers Conference on Broadband Internet Access Technologies Systems & Services*, May 2000.

[38] Lee, K. Adaptive network support for mobile multimedia. *ACM Mobicom'95*, 62–74, Nov. 1995.

[39] Li, J. Visual progressive coding. *SPIE Conf. Visual Communications and Image Processing (VCIP'99)*, SPIE-3653, Jan. 1999.

[40] Li, X., S. Paul, and M. H. Ammar. Layered video multicast with retransmissions (LVMR): Evaluation of hierarchical rate control. *IEEE INFOCOM'98*, 3:1062–72, March 1998.

[41] Liu, H., and M. El Zarki. Performance of H.263 video transmission over wireless channels using hybrid ARQ. *IEEE J. on Selected Areas in Communications* (Dec. 1997), 15(9):1775–86.

[42] Liu, C. L., and J. W. Layland. Scheduling algorithms for multiprogramming in a hard real-time environment. *Journal of ACM* (Jan. 1973), 20(1):46–61.

[43] Liu, Y.-J., and Y.-Q. Zhang. Wavelet-coded image transmission over land mobile radio channels. *IEEE GLOBECOM'92*, 1:235–39, Orlando, FL, Dec. 1992.

[44] Lu, S., K.-W. Lee, and V. Bharghavan. Adaptive service in mobile computing environments. *Fifth International Workshop on Quality of Service (IWQOS'97)*, 25–36, May 1997.

[45] McCanne, S., V. Jacobson, and M. Vetterli. Receiver-driven layered multicast. *ACM SIGCOMM'96*, 117–30, Aug. 1996.

[46] Miao, Z., and A. Ortega. Proxy caching for efficient video services over the Internet. *Packet Video Workshop*, April 1999.

[47] Moura, J., R. S. Jasinschi, H. Shiojiri, and J.-C. Lin. Video over wireless. *IEEE Personal Communications Magazine* (Feb. 1996), 44–54.

[48] Mourad. A. Doubly-striped disk mirroring: reliable storage for video servers. *Multimedia, Tools and Applications* (May 1996), 2:253–72.

[49] Naghshineh, M., and M. Willebeek-LeMair. End-to end QoS provisioning in multimedia wireless/mobile networks using an adaptive framework. *IEEE Communications Magazine* (Nov. 1997), 72–81.

[50] Nanda, S., K. Balachandran, and S. Kumar. Adaptation techniques in wireless packet data services. *IEEE Communications Magazine* (Jan. 2000), 54–64.

[51] Ng, T. S. E., I. Stoica, and H. Zhang. Packet fair queueing algorithms for wireless networks with location-dependent errors. *IEEE INFOCOM'98*, 1103–11, March 1998.

[52] Ortega, A., and K. Ramchandran. Rate-distortion methods for image and video compression. *IEEE Signal Processing Magazine* (Nov. 1998), 74–90.

[53] Ozden, B., R. Rastogi, P. Shenoy, and A. Silberschatz. Fault-tolerant architectures for continuous media servers. *ACM SIGMOD'96*, 79–90, June 1996.

[54] Puri, R., K. W. Lee, K. Ramchandran, and V. Bharghavan. Application of FEC based multiple description coding to Internet video streaming and multicast. *Packet Video Workshop*, May 2000.

[55] Radha, H., and Y. Chen. Fine-granular-scalable video for packet networks. *Packet Video Workshop*, April 1999.

[56] Reddy, A. L., and J. Wyllie. Disk scheduling in a multimedia I/O system. *ACM Multimedia'93*, 289–97.

[57] Reininger, D., R. Izmailov, B. Rajagopalan, M. Ott, and D. Raychaudhuri. Soft QoS control in the WATMnet broadband wireless system. *IEEE Personal Communications Magazine* (Feb. 1999), 34–43.

[58] Rexford, J., S. Sen, and A. Basso. A smoothing proxy service for variable-bit-rate streaming video. *IEEE GLOBECOM'99*, 3:1823–29, Dec. 1999.

[59] Schulzrinne, H., S. Casner, R. Frederick, and V. Jacobson. RTP: a transport protocol for real-time applications. IETF, RFC 1889, Jan. 1996.

[60] Schulzrinne, H. RTP profile for audio and video conferences with minimal control. IETF, RFC 1890, Jan. 1996.

[61] Schulzrinne, H., A. Rao, and R. Lanphier. Real time streaming protocol (RTSP). IETF, RFC 2326, April 1998.

[62] Sen, S., J. Rexford, and D. Towsley. Proxy prefix caching for multimedia streams. *IEEE INFOCOM'99*, 3:1310–19, March 1999.

[63] Shenoy, P., and H. M. Vin. Efficient striping techniques for multimedia file servers. *Performance Evaluation* (Dec. 1999), 38(3–4):175–99.

[64] Silberschatz, A., J. Peterson, and P. Galvin. *Operating System Concepts*. 3rd ed., Reading, MA: Addison Wesley, 1991.

[65] Sodagar, I., H.-J. Lee, P. Hatrack, and Y.-Q. Zhang. Scalable wavelet coding for synthetic/natural hybrid images. *IEEE Trans. on Circuits and Systems for Video Technology* (March 1999), 9(2):244–54.

[66] Steinmetz, R., and K. Nahrstedt. *Multimedia: Computing, Communications and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[67] Sun, H., and W. Kwok. Concealment of damaged block transform coded images using projection onto convex sets. *IEEE Trans. Image Processing* (April 1995), 4:470–77.

[68] Talluri, R. Error-resilience video coding in the ISO MPEG-4 standard. *IEEE Communications Magazine* (June 1998), 112–19.

[69] Tan, W., and A. Zakhor. Real-time Internet video using error resilient scalable compression and TCP-friendly transport protocol. *IEEE Trans. Multimedia* (June 1999), 1(2):172–86.

[70] Tanenbaum, A. S. *Modern Operating Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[71] Taubman, D., and A. Zakhor. A common framework for rate and distortion based scaling of highly scalable compressed video. *IEEE Trans. on Circuits and Systems for Video Technology* (Aug. 1996), 6(4):329–54.

[72] Tewari, R., D. M. Dias, W. Kish, and H. Vin. Design and performance trade-offs in clustered video servers. *IEEE International Conference on Multimedia Computing and Systems* (June 1996), 144–50.

[73] Tham, J. Y., S. Ranganath, and A. A. Kassim. Highly scalable wavelet-based video codec for very low bit-rate environment. *IEEE J. Select. Areas Commun.* (Jan. 1998), 16(1):12–27.

[74] Turletti, T., and C. Huitema. Videoconferencing on the Internet. *IEEE/ACM Trans. on Networking* (June 1996), 4(3):340–51.

[75] Vetro, A., H. Sun, and Y. Wang. Object-based transcoding for scalable quality of service. *IEEE ISCAS'2000*, 4:17–20, Geneva, May 28–31, 2000.

[76] Vin, H. M., P. Goyal, A. Goyal, and A. Goyal. A statistical admission control algorithm for multimedia servers. *ACM Multimedia'94*, 33–40, Oct. 1994.

[77] Wang, X., and H. Schulzrinne. Comparison of adaptive Internet multimedia applications. *IEICE Trans. Commun.* (June 1999), E82-B(6):806–18.

[78] Wang, Y., Q.-F. Zhu, and L. Shaw. Maximally smooth image recovery in transform coding. *IEEE Trans. Commun.* (Oct. 1993), 41(10): 1544–51.

[79] Wang, Y., M. T. Orchard, and A. R. Reibman. Multiple description image coding for noisy channels by pairing transform coefficients. *IEEE Workshop on Multimedia Signal Processing*, 419–24, June 1997.

[80] Wang, Y., and Q.-F. Zhu. Error control and concealment for video communication: a review. *Proceedings of the IEEE* (May 1998), 86(5):974–97.

[81] Wu, D., et al. On end-to-end architecture for transporting MPEG-4 video over the Internet. Forthcoming in *IEEE Trans. on Circuits and Systems for Video Technology.*

[82] Wu, D., Y. T. Hou, Y.-Q. Zhang, W. Zhu, and H. J. Chao. Adaptive QoS control for MPEG-4 video communication over wireless channels. *IEEE ISCAS'2000,* 1:48–51, Geneva, May 28–31, 2000.

[83] Wu, D., et al. An end-to-end approach for optimal mode selection in Internet video communication: Theory and application. *IEEE Journal on Selected Areas in Communications* (June 2000), 18(6):977–95.

[84] Wu, D., Y. T. Hou, and Y.-Q. Zhang. Scalable video transport over wireless IP networks, in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communication (PIMRC'2000),* 2:1185–91, London, Sept. 18–21, 2000.

[85] Wu, D., Y. T. Hou, and Y.-Q. Zhang. Scalable video coding and transport over broadband wireless networks. Forthcoming in *Proceedings of the IEEE.*

[86] Xu, X. R., A. C. Myers, H. Zhang, and R. Yavatkar. Resilient multicast support for continuous-media applications. *IEEE 7th Int Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'97),* 183–94.

[87] Yeadon, N., F. Garcia, D. Hutchison, and D. Shepherd. Filters: QoS support mechanisms for multipeer communications. *IEEE Journal on Selected Areas in Communications* (Sept. 1996), 14(7):1245–62.

[88] Yu, P. S., M. S. Chen, and D. D. Kandlur. Grouped sweeping scheduling for DASD-based multimedia storage management. *ACM/Springer Multimedia Systems* (1993), 1(3):99–109.

[89] Zhang, Q., and S. A. Kassam. Hybrid ARQ with selective combining for fading channels. *IEEE Journal on Selected Areas in Communications* (May 1999), 17(5):867–80.

[90] Zhang, Z.-L., S. Nelakuditi, R. Aggarwa, and R. P. Tsang. Efficient selective frame discard algorithms for stored video delivery over resource constrained networks. *IEEE INFOCOM'99,* 2:472–79, March 1999.

[91] Zhang, Z.-L., Y. Wang, D. H. C. Du, and D. Su. Video staging: A proxy-server-based approach to end-to-end video delivery over wide-area networks, *IEEE/ACM Transactions on Networking* (Aug. 2000), 8(4):429–42.

附录 A　空时梯度的确定

为了实现本书给出的许多最优化问题所需要的各种梯度下降法,我们必须能够确定视频信号的空间和时间梯度。这个附录给出几种用差分算子近似连续梯度运算的方法。

A.1　一阶和二阶梯度

近似沿特定方向的一阶梯度的最简单方法是利用该方向上两个像素值之差,由此得到:

$$\left.\frac{\partial \psi}{\partial x}\right|_{(x,y,t)} = \psi(x,y,t) - \psi(x-1,y,t)$$

$$\left.\frac{\partial \psi}{\partial y}\right|_{(x,y,t)} = \psi(x,y,t) - \psi(x,y-1,t)$$

$$\left.\frac{\partial \psi}{\partial t}\right|_{(x,y,t)} = \psi(x,y,t) - \psi(x,y,t-1)$$

用这个近似,我们可以进一步导出二阶空间梯度算子的近似:

$$\begin{aligned}
\left.\frac{\partial^2 \psi}{\partial x^2}\right|_{(x,y,t)} &= \left.\frac{\partial \psi}{\partial x}\right|_{(x+1,y,t)} - \left.\frac{\partial \psi}{\partial x}\right|_{(x,y,t)} \\
&= \psi(x+1,y,t) - 2\psi(x,y,t) + \psi(x-1,y,t) \\
\left.\frac{\partial^2 \psi}{\partial y^2}\right|_{(x,y,t)} &= \left.\frac{\partial \psi}{\partial x}\right|_{(x,y+1,t)} - \left.\frac{\partial \psi}{\partial x}\right|_{(x,y,t)} \\
&= \psi(x,y+1,t) - 2\psi(x,y,t) + \psi(x,y-1,t) \\
\left.\frac{\partial^2 \psi}{\partial xy}\right|_{(x,y,t)} &= \left.\frac{\partial \psi}{\partial x}\right|_{(x+1,y+1,t)} - \left.\frac{\partial \psi}{\partial x}\right|_{(x+1,y,t)} \\
&= \psi(x+1,y+1,t) - \psi(x+1,y,t) - \psi(x,y+1,t) + \psi(x,y,t) \\
\left.\frac{\partial^2 \psi}{\partial t^2}\right|_{(x,y,t)} &= \left.\frac{\partial \psi}{\partial t}\right|_{(x,y,t+1)} - \left.\frac{\partial \psi}{\partial t}\right|_{(x,y,t)} \\
&= \psi(x,y,t+1) - 2\psi(x,y,t) + \psi(x,y,t-1)
\end{aligned}$$

A.2　索贝尔算子

上一节给出的数字近似对视频数据中的噪声很敏感。而且,有些算子是不对称的。在图像处理中,一阶空间梯度经常用索贝尔(Sobel)算子近似,它在沿梯度方向取中心差分之前,沿切线方向进行平滑,特别地:

$$\begin{aligned}
\left.\frac{\partial \psi}{\partial x}\right|_{(x,y,t)} = &\psi(x+1,y-1,t) + 2\psi(x+1,y,t) + \psi(x+1,y+1,t) \\
&- \psi(x-1,y-1,t) - 2\psi(x-1,y,t) - \psi(x-1,y+1,t)
\end{aligned}$$

$$\left.\frac{\partial \psi}{\partial y}\right|_{(x,y,t)} = \psi(x-1,y+1,t) + 2\psi(x,y+1,t) + \psi(x+1,y+1,t)$$

$$-\psi(x-1,y-1,t)-2\psi(x,y-1,t)-\psi(x+1,y-1,t)$$

注意，这些运算可以用一个 3×3 滤波器简单地实现。

A.3 高斯差分滤波器

一阶(或二阶)梯度的一个更具有噪声健壮性的实现是，首先使原始数据通过一个高斯平滑滤波器，然后对平滑后的信号应用一阶(或二阶)差分算子。由于平滑算子(卷积算子)与差分算子是可互换的，这个运算等价于将原始信号与一个滤波器进行卷积，这个滤波器是高斯函数的一阶(或二阶)梯度。为了把这个滤波器应用于数字信号，上述连续域的滤波器必须进行采样和截断。所得到的高斯一阶梯度(或高斯的差分)称为 DoG 滤波器，而高斯的二阶梯度或拉普拉斯算子称为 LoG 滤波器。

令 $g(x,y) = \exp(-(x^2+y^2/2\sigma^2))$代表高斯函数，则

$$\frac{\partial g}{\partial x} = -\frac{x}{\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

$$\frac{\partial g}{\partial y} = -\frac{y}{\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

$$\frac{\partial^2 g}{\partial x^2} = -\frac{1}{\sigma^2}\left(1-\frac{x^2}{\sigma^2}\right)\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

$$\frac{\partial^2 g}{\partial y^2} = -\frac{1}{\sigma^2}\left(1-\frac{y^2}{\sigma^2}\right)\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

$$\frac{\partial^2 g}{\partial xy} = \frac{xy}{\sigma^4}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

显然，所有这些滤波器是可分离的。为得到任何一个这种函数的数字近似，可以把它的长度截取为 $2\sim3\sigma$，并对截断的函数在整数采样点上进行采样。参数 σ 应该以像素为单位定义，而且应该基于数据中出现的噪声量选择。通常的近似是设置 $\sigma=1$ 像素，在 2σ 截断该函数。得到的滤波器是 5×5 阶滤波器，如下所示：

$$[\mathbf{G}_x] = \begin{bmatrix} 0.0366 & 0.0821 & 0 & -0.0821 & -0.0366 \\ 0.1642 & 0.3679 & 0 & -0.3679 & -0.1642 \\ 0.2707 & 0.6065 & 0 & -0.6065 & -0.2707 \\ 0.1642 & 0.3679 & 0 & -0.3679 & -0.1642 \\ 0.0366 & 0.0821 & 0 & -0.0821 & -0.0366 \end{bmatrix}$$

$$[\mathbf{G}_{xx}] = \begin{bmatrix} 0.0549 & 0 & -0.1353 & 0 & 0.0549 \\ 0.2463 & 0 & -0.6065 & 0 & 0.2463 \\ 0.4060 & 0 & -1.0000 & 0 & 0.4060 \\ 0.2463 & 0 & -0.6065 & 0 & 0.2463 \\ 0.0549 & 0 & -0.1353 & 0 & 0.0549 \end{bmatrix}$$

$$
[\mathbf{G}_{xy}] = \begin{bmatrix} 0.0733 & 0.1642 & 0 & -0.1642 & -0.0733 \\ 0.1642 & 0.3679 & 0 & -0.3679 & -0.1642 \\ 0 & 0 & 0 & 0 & 0 \\ -0.1642 & -0.3679 & 0 & 0.3679 & 0.1642 \\ -0.0733 & -0.1642 & 0 & 0.1642 & 0.0733 \end{bmatrix}
$$

$$
[\mathbf{G}_y] = [\mathbf{G}_x]^T; \qquad [\mathbf{G}_{yy}] = [\mathbf{G}_{xx}]^T
$$

附录 B　梯度下降法

本附录考察一些用于最小化目标函数的基本梯度下降法。我们考虑当函数是 K 维多变量的一般情况。

令 $J(\mathbf{x})$代表目标函数,其中 $\mathbf{x} = [x_1, x_2, \cdots, x_k]^T$是一个 K 维矢量。梯度下降法是迭代的方法,它确定达到 $J(\mathbf{x})$ 的最小值的点 $\mathbf{x}^*$。

B.1　一阶梯度下降法

令 $\mathbf{x}^{(l)}$代表第 l 次迭代的解,一阶梯度下降法正比于目标函数的一阶梯度更新这个解,即:

$$\mathbf{x}^{(l+1)} = \mathbf{x}^{(l)} - \alpha \frac{\partial J}{\partial \mathbf{x}}\bigg|_{\mathbf{x}^{(l)}} \tag{B.1.1}$$

一阶梯度下降法潜在的基本原理是$\partial J/\partial \mathbf{x}$ 代表函数 $J(\mathbf{x})$增长最快的方向,从而 $-\partial J/\partial \mathbf{x}$ 就是 $J(\mathbf{x})$下降最快的方向。

常数 α 称为步长。它必须适当地选取,以保证迭代过程收敛于期望的最小点 $\mathbf{x}^*$。如果 α 太大,解会在 $\mathbf{x}^*$ 附近振荡;而如果 α 太小,则需要进行太多的迭代才能收敛。显然,如果函数 $J(\mathbf{x})$具有多个局部最小,那么收敛的解将是距离初始解最近的那个局部最小。因此,选择一个适当的初始解是非常重要的。为了得到全局最小,我们也可以从几个距离足够远的不同的初始解,然后从每个初始解运行这个迭代方案,通过比较由不同初始解得到的所有局部最小处的函数值,取其中具有最小值的一个。然而,不能保证它一定是全局最小。更复杂的方法是模拟退火(simulated annealing),它允许迭代过程跳出局部最小[1]。

B.2　最陡下降法

在上述方法中,步长 α 设为一个小的常数。为加速收敛,我们也可以在每一次新的迭代中改变 α,使 $J(\mathbf{x})$达到最大的降低。这是可能的,因为对固定的 $\mathbf{x}^{(l)}$ 和$(\partial J/\partial \mathbf{x})|_{\mathbf{x}^{(l)}}$,新函数值 $J(\mathbf{x}^{(l+1)})$是 α 的标量函数。可以采用数值搜索法来确定每一次迭代的最优步长。采用这样的可变步长的一阶梯度下降法称为最陡下降法。与固定步长算法相比,这种方法法只要经过较少的迭代就能收敛,但是每一次迭代将需要更多的时间来确定最优步长。

B.3　牛顿法

函数达到局部最小的必要条件是它在该点的梯度为零。给定上一次迭代的解 $\mathbf{x}^{(l)}$,此处的梯度还不是零,一个确定更新的方法是要求更新位置处的梯度为零,即:

$$\frac{\partial J}{\partial \mathbf{x}}\bigg|_{\mathbf{x}^{(l)}+\Delta\mathbf{x}} = 0 \tag{B.3.1}$$

上述梯度函数利用泰勒公式展开至第一项,我们得到:

$$\left.\frac{\partial J}{\partial \mathbf{x}}\right|_{\mathbf{x}^{(l)}} + [\mathbf{H}(\mathbf{x}^{(l)})]\Delta\mathbf{x} = 0 \tag{B.3.2}$$

其中

$$[\mathbf{H}(\mathbf{x})] = \frac{\partial^2 J}{\partial \mathbf{x}^2} = \begin{bmatrix} \frac{\partial^2 J}{\partial x_1 \partial x_1} & \frac{\partial^2 J}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 J}{\partial x_1 \partial x_K} \\ \frac{\partial^2 J}{\partial x_2 \partial x_1} & \frac{\partial^2 J}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 J}{\partial x_2 \partial x_K} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 J}{\partial x_K \partial x_1} & \frac{\partial^2 J}{\partial x_K \partial x_2} & \cdots & \frac{\partial^2 J}{\partial x_K \partial x_K} \end{bmatrix}$$

它是 $J(\mathbf{x})$的二阶导数,称为 Hessian 矩阵。解方程(B.3.2),我们得到:

$$\Delta\mathbf{x} = -[\mathbf{H}(\mathbf{x}^{(l)})]^{-1}\left.\frac{\partial J}{\partial \mathbf{x}}\right|_{\mathbf{x}^{(l)}}$$

或

$$\mathbf{x}^{(l+1)} = \mathbf{x}^{(l)} - [\mathbf{H}(\mathbf{x}^{(l)})]^{-1}\left.\frac{\partial J}{\partial \mathbf{x}}\right|_{\mathbf{x}^{(l)}}$$

上述迭代算法称为牛顿法,除一阶导数外它还需要二阶导数。牛顿法的收敛速度很快,但是每一步迭代需要进行大量的计算,且这个算法并不保证收敛。二阶导数的使用也使这个算法易于产生噪声和数值误差。

实际上,更经常采用如下的更新:

$$\mathbf{x}^{(l+1)} = \mathbf{x}^{(l)} - \alpha[\mathbf{H}(\mathbf{x}^{(l)})]^{-1}\left.\frac{\partial J}{\partial \mathbf{x}}\right|_{\mathbf{x}^{(l)}} \tag{B.3.3}$$

其中,步长 α 是小于 1 的常数。它必须适当地选取,以达到确保收敛和收敛速度之间的折中。

B.4 牛顿-拉夫森算法

若目标函数是误差的平方和,即:

$$J(\mathbf{x}) = \frac{1}{2}\sum_k e_k^2(\mathbf{x}) \tag{B.4.1}$$

我们有:

$$\frac{\partial J}{\partial \mathbf{x}} = \sum \frac{\partial e_k}{\partial \mathbf{x}} e_k(\mathbf{x}) \tag{B.4.2}$$

和

$$[\mathbf{H}] = \frac{\partial^2 J}{\partial \mathbf{x}^2} = \sum \frac{\partial e_k}{\partial \mathbf{x}}\left(\frac{\partial e_k}{\partial \mathbf{x}}\right)^T + \frac{\partial^2 e_k}{\partial \mathbf{x}^2} e_k(\mathbf{x}) \approx \sum \frac{\partial e_k}{\partial \mathbf{x}}\left(\frac{\partial e_k}{\partial \mathbf{x}}\right)^T \tag{B.4.3}$$

最后的近似是由假设二阶导数的项远小于一阶导数的项得到的。在公式(B.3.3)中采用这个近似就得到牛顿-拉夫森算法。由于实际中遇到的几乎所有误差最小化问题都用误差平方和作为目标函数,所以牛顿-拉夫森算法被广泛采用。

B.5 文献目录

[1] Van Laarhoven, P.J.M., and E.H.Aarts. *Simulated Annealing: Theory and Applications*. Dordrecht, The Netherlands: Reidel, 1987.

附录 C 缩写词汇表

2-D	二维
3-D	三维
3DTV	三维电视
K-D	K 维
AAC	先进音频编码器(MPEG-2 音频编码器)
ABR	可用比特率(ATM 网络的一种服务类型)
AC	交流(指相应于非零频率的变换系数)
ACE	先进的编码效率(一种 MPEG-4 视频编码类)
A/D	模拟到数字
ADSL	非对称数字用户环路
ALF	应用层成帧
ARQ	自动重传请求
ART	角射线变换
ARTS	先进的实时简单(一种 MPEG-4 视频编码类)
ASP	应用服务供应商
AU	动作单元(用于 FACS)
AV	音视频
ATM	异步传输模式
BAB	二进制阿尔法块(一种 MPEG-4 形状编码单元)
BDP	身体定义参数(用于 MPEG-4)
BAP	身体动画参数(用于 MPEG-4)
BER	误码率
BIFS	二进制格式场景(MPEG-4 场景建模语言)
B-ISDN	宽带 ISDN
BMA	块匹配算法
bpp	比特/像素
bps	比特/秒
BT.601	ITU-R 制订的一种数字视频格式,以前称为 CCIR601
CAC	呼叫接纳控制
CBP	编码块模式(为 MPEG-1/2/4 标准中的每一宏块指定的)
CBR	恒定比特率(在 ATM 网络中的一种服务类型)
CCD	电荷耦合器件
CCIF	国际电话咨询委员会
CCIR	国际无线电咨询委员会

CCIR601	见 BT.601
CCIT	国际电报咨询委员会
CCITT	国际电话和电报咨询委员会(更名为 ITU-T)
CDMA	码分多址
CELP	码激励线性预测(一种语音编码方法)
CIE	国际照明委员会(彩色专家的一个国际组织)
CIF	普通中间媒体格式(ITU-T 制订的一种数字视频格式)
CIFQ	独立于信道条件的公平排队
CLR	单元丢失率(在 ATM 网络中)
CMY	用青、品、黄作为基色的彩色坐标
CMYK	用青、品、黄和黑作为基色的彩色坐标
codec	编码器和解码器
cpd	周期/度,在水平或垂直方向上的空间频率单位
CPS	约束参数集
CRT	阴极射线管
CSFT	连续空间傅里叶变换
CSP	内容服务供应商
CSS	曲率尺度空间
DAI	传送应用接口
DAVIC	数字音视频委员会
DBMA	可变形块匹配算法
DBS	直接广播卫星
DC	直流(指相应于零频的变换系数)
DCP	视差补偿预测
DCT	离散余弦变换
DFD	移位帧差
DFT	离散傅里叶变换
DSFT	离散空间傅里叶变换
DTV	数字电视
DMIF	传送多媒体集成框架
DMV	差分运动矢量(MPEG-2 视频中定义的运动校正矢量)
DPCM	差分脉冲编码调制
DSL	数字用户环路
DSP	数字信号处理器
DV	位差矢量
DVD	数字视频光盘
DVTR	数字视频磁带录像机
DWT	离散小波变换
EBMA	穷尽搜索块匹配算法

EDF	最早截止时间优先
EDGE	增强数据率的 GSM 进化
EC	Colomb 指数(一种概率分布)
EOB	块结束(用于编码 DCT 块的符号)
EREC	差错复原熵编码
EZW	嵌入零树小波(一种基于小波的编码方法)
F2D	柔性的二维
F3D	柔性的三维
FACS	面部动作编码系统
FAP	脸动画参数(用于 MPEG-4)
FC-AL	光纤通道仲裁环路
FDDI	光纤分配数据接口
FDP	脸定义参数(用于 MPEG-4)
FEC	前向纠错
FEM	有限元法
FFT	快速傅里叶变换
FGS	细间隔度可分级性
FIFO	先进先出
fps	帧/秒、帧率或时间采样频率的单位(也表示为 Hz)
GGD	广义高斯分布
GMF	高斯-马尔可夫场
GOB	块组(视频编码标准中的一个术语)
GOP	图像组(视频编码标准中的一个术语)
GPRS	通用包无线电业务
GR	Golomb-Rice(一种概率分布)
GSM	全球移动通信系统
GSS	分组扫描调度
GVOP	视频对象平面组(MPEG-4 中的一个术语)
HBMA	分层块匹配算法
HDTV	高清晰度电视
HFM-ME	分层的特征匹配运动估计方法
HIS	使用色调、饱和度和强度的彩色坐标(又称为 HSV,此处 V 代表值)
HTTP	超文本传输协议
HVS	人类视觉系统
I3D	不完全的三维
IAR	图像幅型比
IEC	国际电工委员会
IETF	因特网工程任务组
i.i.d	独立和恒等分布

IP	因特网协议
ISD	独立段解码
ISDN	综合业务数字网(通常指窄带 ISDN)
ISA	国家标准化协会国际联盟
ISO	国际标准化组织
ISP	因特网服务供应商
ITU	国际电信联盟
ITU-R	国际电信联盟无线电部
ITU-T	国际电信联盟电信部
JPEG	联合图片专家组
KBASC	基于知识的分析合成编码
kbps	千比特/秒
KLT	Karhuner-Loeve 变换
LAN	局域网
LC	分层编码
LCD	液晶显示器
LDU	逻辑数据单元
LMDS	本地多点分配业务
LMMSE	线性最小均方误差
lpf	行/帧
LSI	线性移不变
LTI	线性时不变
MAD	平均绝对差
MAN	城域网
MAP	最大后验
MB	宏块(ITU 和 MPEG 视频编码标准中的编码单位)
Mbps	兆比特/秒
MC	模型一致性
MCI	运动补偿内插
MCP	运动补偿预测
MCU	多点控制单元
MDC	多描述编码
MDL	最小描述长度(一种信源编码方法)
MDTC	多描述变换编码
MF	模型失效
MMDS	多信道多点分配业务
MMSE	最小均方误差
MOS	平均意见分
MPEG	运动图像专家组

MQUANT	视频编码标准中宏块的量化步长
ms	毫秒
MSB	最高有效位
MSE	均方误差
MSR	最大平滑恢复(一种错误隐藏方法)
MTF	调制传输函数
MTYPE	视频编码标准中宏块的编码模式
MTU	最大传输单元(因特网数据传输中最大的数据包尺寸)
MV	运动矢量
MVD	视频编码标准中差分编码的运动矢量
MZTE	多级零树熵编码(一种基于小波的编码方法)
nm	纳米
NTSC	用于大多数北美国家的模拟电视制式,由美国国家电视系统委员会制订
OBASC	基于对象的分析合成编码
OBMC	重叠块运动补偿
OD	对象描述子(一个 MPEG-4 实体)
OF	光流
OTS	基于对象的时间可分级性
PAL	逐行倒相,大多数西欧国家采用的模拟电视制式
PAR	像素幅型比
PCF	相位相关函数
PCM	脉冲编码调制
PCT	成对相关变换
pdf	概率密度函数
pmf	概率质量函数
POCS	凸集投影
POTS	简单的老式电话系统
PS	节目流
PSD	功率谱密度
PSNR	峰值信噪比
PSTN	公共交换电话网
PTS	演播时间戳
QAM	正交幅度调制
QCIF	四分之一普通中间媒体格式
QoS	服务质量
QP	量化参数(控制量化器步长的参数)
R3D	刚性的三维
RCPC	码率兼容删除卷积码
RD	率失真

READ	相对单元地址指定
RGB	用红、绿、蓝作为基色的彩色坐标
RISC	精简指令集计算机
RPS	参考图像选择
RS	Reed-Solomon(一种类型的信道纠错码)
RTCP	实时控制协议
RTP	实时传输协议
RTT	往返时间
RTSP	实时流协议
RV	随机变量
RVLC	可逆的可变长编码
SA-DCT	形状自适应 DCT
SA-DWT	形状自适应 DWT
SA-ZTE	形状自适应 ZTE
SAN	存储区域网
SAP	话路通告协议
SCSI	小型计算机系统接口
SDP	话路描述协议
SDTV	标准清晰度电视
SECAM	顺序彩色与存储,大多数东欧国家采用的模拟电视制式
SG	研究组(ITU 的一个下属机构)
SIP	话路启动协议
SL	同步层
SMPTE	电影与电视工程师协会
SNR	信噪比
SPIHT	分层树集分割(一种基于小波的编码算法)
SQ	标量量化
SSFT	采样空间傅里叶变换
SSP	流同步协议
SSRC	同步信源识别符
STC	系统时钟
STF	截断符号的特征(用于 HFM-ME 的特征)
SVD	奇异值分解
S-VHS	超级 VHS,VHS 的加强版
TC	变换编码
TCP	传输控制协议
TCQ	格形编码的量化
TDMA	时分多址
TFT	薄膜晶体管

TS	传输流
TTS	文本到语音
TTSI	TTS 接口
TV	电视
UBR	未指定比特率(ATM 网络中的一种业务类型)
UDP	用户数据包协议
URL	通用资源定位器
UMTS	通用移动通信系统
VBR	可变比特率(ATM 网络中的一种业务类型)
VCD	视频压缩光盘
VCR	视频盒式磁带录像机
VHS	模拟视频磁带录像格式
VLC	可变长编码
VLD	可变长解码
VO	视频对象(MPEG-4 中的术语)
VOL	视频对象层(MPEG-4 中的术语)
VOP	视频对象平面(MPEG-4 中的术语)
VRML	虚拟现实模拟语言
VQ	矢量量化
VSB	残留边带调制
VTR	视频磁带录像机
WAN	广域网
WP	工作组(ITU 的一个下属机构)
YCbCr	用于大多数数字视频格式的彩色坐标,包含一个亮度(Y)和两个色差信号(Cb 和 Cr)分量
YIQ	用于 NTSC 制式的彩色坐标,包含一个亮度(Y)和两个色度分量 I 和 Q
YUV	用于 PAL 和 SECAM 制式的彩色坐标,包含亮度(Y)和两个色度分量 U 和 V
ZTE	零树熵编码(一种基于小波的编码方法)

附录D　部分习题答案

第1章

1.2:对于照明光,红 + 绿 + 蓝 = 白,红 + 绿 = 黄。

1.3:对于颜料,红 + 绿 + 蓝 = 黑,红 + 绿 = 蓝。

1.4:用公式(1.4.4)和(1.4.2)计算。

对 YIQ:(a) (1,0,0),(b) (0.587, -0.274, -0.523),(c) (0.886,0.321, -0.311),(d) (0.701, -0.596, -0.211)。

对 YUV:(a) (1,0,0),(b) (0.587, -0.289, -0.515),(c) (0.886, -0.436,0.100),(d) (0.701,0.147, -0.615)。

1.5:用公式(1.5.3)计算。(a) (235,128,128),(b) (144,54,34),(c) (210,16,146),(d) (170,166,16)。

1.9:对于 PAL 和 SECAM,亮度带宽都是 5.04 MHz。

第2章

2.1:

$$H(f_x,f_y,f_t) = \frac{\sin(\pi f_x T_x)}{\pi f_x T_x}\frac{\sin(\pi f_y T_y)}{\pi f_y T_y}\frac{\sin(\pi f_t \Delta_e)}{\pi f_t \Delta_e}\exp(-\mathrm{j}\pi f_t \Delta_e)$$

2.2:投影图像的 CSFT 是:

$$\Psi(f_x,f_y,f_t) = B^2\frac{\sin(\pi f_x B)}{\pi f_x B}\frac{\sin(\pi f_y B)}{\pi f_y B}\delta(f_t + f_x v_x)$$

摄取信号的 CSFT 是:

$$\begin{aligned}\Psi_c(f_x,f_y,f_t) &= \Psi(f_x,f_y,f_t)H(f_x,f_y,f_t)\\ &= B^2\frac{\sin(\pi f_x B)}{\pi f_x B}\frac{\sin(\pi f_y B)}{\pi f_y B}\frac{\sin(\pi f_x T_x)}{\pi f_x T_x}\frac{\sin(\pi f_y T_y)}{\pi f_y T_y}\frac{\sin(\pi f_x v_x \Delta_e)}{\pi f_x v_x \Delta_e}\\ &\quad\times\exp(\mathrm{j}\pi f_x v_x \Delta_e)\delta(f_t + f_x v_x)\end{aligned}$$

感觉上,摄取的图像是一个水平运动的模糊方块。

2.5:距离 d = 3,1,5 米观看时,角频率分别为 f_θ = 5.29,1.88,8.75 cpd。当 f_θ 为 3~5 cpd 时 HVS 最敏感,因此 d = 3 米比其他两种安排更好。

2.6:当(v_x,v_y) = (1,1),(4,-3),(4,0),(0,1)时,时间频率分别为 f_t = 7,0,12,4 Hz。

2.7:当眼睛精确地跟踪物体运动时,视网膜上的时间频率 f'_t = 0。对于固定的眼速度(2,2),f'_t 分别为 7,14,2,10 Hz。

第3章

3.1:(a)点阵 A 的沃罗纳晶格是六边形,采样密度是 $1/2\sqrt{3}$。点阵 B 的沃罗纳晶格是菱

形，采样密度是 1/4。(b)点阵 A 的反商点阵的基矢量是$[1/\sqrt{3},0]$，$[-1/2\sqrt{3},1/2]$，它的沃罗纳晶格仍然是一个六边形，但是与原始点阵的方位不同，采样密度是 $2\sqrt{3}$。点阵 B 的反商点阵的基矢量是$[\sqrt{2}/4,-\sqrt{2}/4]$，$[0,\sqrt{2}/2]$，它的沃罗纳晶格仍然是一个菱形，采样密度是 4。(c)令 r_{max}代表采样时不发生混叠的频谱的最大半径，则 $r_{A,max}=1/2\sqrt{3}$，$r_{B,max}=1/4$。因为 $r_{A,max}>r_{B,max}$，所以点阵 A 更好。

3.7：(a) $\Psi_c(f_1,f_2)=\sqrt{2\pi}\exp(-2\pi^2(f_1^2+f_2^2))$，(b)$\alpha=1.43$

(c)

$$\psi_s(n_1,n_2)=\frac{1}{\sqrt{2\pi}}\exp(-(n_1^2+n_1n_2+n_2^2)\alpha^2/2)$$

$$\Psi_s(f_1,f_2)=\frac{4\pi}{\sqrt{3}\alpha}\sum_{m_1,m_2\in Z}\exp\left(-2\pi^2\left(\left(f_1-\frac{2m_1-m_2}{\sqrt{3}\alpha}\right)^2+\left(f_2-\frac{m^2}{\alpha}\right)^2\right)\right)$$

对于(d)和(e)，见 MATLAB 手稿 prob3_7.m(可查询网址 www.prenhall.com/wang)。

3.9：在所有的情况下，空间采样率是 $f_{s,x}=f_{s,y}=4$ 样点/米。以样点/秒或 Hz 为单位的时间采样率 $f_{s,t}$是(a)0，(b)8，(c)4。

3.10：表观空间频率是 $f_x=2$，$f_y=-2$，这个条看起来是静止的。

3.13：

$$H_p(f_x,f_y,f_t)=\frac{\sin(\pi f_xT_x)}{\pi f_xT_x}\frac{\sin(\pi f_yT_y)}{\pi f_yT_y}\frac{\sin(\pi f_t\Delta_e)}{\pi f_t\Delta_e}\exp(-j\pi f_t\Delta_e)$$

第 4 章

4.1：顶场的去隔行滤波器是：

$$h_1(x,y,t)=\begin{cases}1/2, & t=0,y=\pm\Delta y;\\ 1/4, & t=-\Delta_t,y=0;\\ -1/8, & t=-\Delta_t,y=\pm2\Delta_y;\\ 0, & \text{其他}\end{cases}$$

SSFT 是：$H_1(f_y,f_t)=\cos(2\pi f_y\Delta_y)+\frac{1}{4}(1-\cos(4\pi f_y\Delta_y))e^{j2\pi f_t\Delta t}$，

底场的去隔行滤波器是：$h_2(x,y,t)=h_1(x,y,-t)$，$H_2(f_y,f_t)=H_1(f_y,-f_t)$。

4.2：去隔行格栅上的滤波器为：

$$h[n\Delta_y,0]=\left[-\frac{1}{32},0,\frac{9}{32},1,\frac{9}{32},0,-\frac{1}{32}\right],n=-3,\cdots,3,h[0,\Delta_t]=\frac{1}{2}$$

其中 $\Delta_t=1/60$ s，$\Delta_y=\frac{1}{525}$像高。

DSFT 为： $H_{DSFT}(f_y,f_t)=1+\frac{18}{32}\cos2\pi f_y-\frac{2}{32}\cos6\pi f_y+\frac{1}{2}\exp(-j2\pi f_t)$

SSFT 为：$H_{SSFT}(f_y,f_t)=H_{DSFT}(\Delta_y f_y,\Delta_t f_t)$

4.5：(b)中间点阵上的等效滤波器（120 fps，逐行）是：$h(0,m\Delta_t)=1,m=0,1,2,3,4$。

SSFT 为：$H_{SSFT}(f_y,f_t)=(1+2\cos 2\pi\Delta_t f_t+2\cos 4\pi\Delta_t f_t)\exp(-j4\pi\Delta_t f_t)$

4.6：(b)场 1 的预滤波器（垂直方向）是：

$$h_1[n2\Delta_y]=\frac{1}{137}[-29,0,88,138,88,0,-29],n=-3,\cdots,3$$

它的 DSFT 是：$H_1(f_y)=1+\frac{176}{138}\cos 2\pi f_y-\frac{58}{138}\cos 6\pi f_y$。

场 2 的内插滤波器是： $h_2[n\delta_y]=\frac{1}{16}[1,0,7,16,7,0,1],n=-3,\cdots,3$

其 DSFT 是：$H_2(f_y)=1+\frac{14}{16}\cos 2\pi f_y+\frac{2}{16}\cos 6\pi f_y$。

总的垂直-时间滤波器是：

$$h_3(n\Delta_y,0)=\left[-\frac{29}{138},0,0,\frac{1}{16},\frac{88}{138},\frac{7}{16},1,\frac{7}{16},\frac{88}{138},\frac{1}{16},0,0,-\frac{29}{138}\right]n=-6,\cdots,6,$$

DSFT 是：$H_3(f_y,f_t)=1+\frac{14}{16}\cos 2\pi f_y+\frac{176}{138}\cos 4\pi f_y+\frac{2}{16}\cos 6\pi f_y-\frac{58}{138}\cos 12\pi f_y$，

相应的 SSFT 分别是：$H_{1,SSFT}(f_y)=H_1(2\Delta_y f_y)$，$H_{2,SSFT}(f_y)=H_2(\Delta_y f_y)$和

$H_{3,SSFT}(f_y,f_t)=H_3(\Delta_y f_y,\Delta_t f_t)$。

4.7：(b)对于场 1，

$$h_1[n2\Delta_y]=[1/2,1,1/2],n=-1,0,1$$

对于场 2，

$$h_2[n\Delta_y]=\left[\frac{1}{4},0,\frac{3}{4},1,\frac{3}{4},0,\frac{1}{4}\right],n=-3,\cdots,3$$

总的滤波器是：

$$h_3[n\Delta_y,0]=\left[\frac{1}{4},\frac{1}{2},\frac{3}{4},1,\frac{3}{4},\frac{1}{2},\frac{1}{4}\right],n=-3,\cdots,3$$

SSFT 是：

$$H_3(f_y,f_t)=1+\frac{3}{2}\cos 2\pi\Delta_y f_y+\cos 4\pi\Delta_y f_y+\frac{1}{2}\cos 6\pi\Delta_y f_y$$

第 5 章

5.1：由三维运动(X,Y,Z)到(X,Y,Z')所引起的 x 轴方向上的运动是 $\Delta x=FX(Z-Z')/ZZ'$。每个图像像素的宽度是 $\Delta W=W/N_x$，其中 $W=1.33''$，$N_x=352$。令 $\Delta x=\Delta W$，则有 $\Delta Z_x=Z'-Z=-Z/(1+XN_x/W)$。类似地，$\Delta Z_y=-Z/(1+YN_y/H)$。

5.5：仿射参数可以通过解下列方程求得：

$$\begin{bmatrix}1 & x_1 & y_1\\ 1 & x_2 & y_2\\ 1 & x_3 & y_3\end{bmatrix}\begin{bmatrix}a_0\\ a_1\\ a_2\end{bmatrix}=\begin{bmatrix}d_{x,1}\\ d_{x,2}\\ d_{x,3}\end{bmatrix},\begin{bmatrix}1 & x_1 & y_1\\ 1 & x_2 & y_2\\ 1 & x_3 & y_3\end{bmatrix}\begin{bmatrix}b_0\\ b_1\\ b_2\end{bmatrix}=\begin{bmatrix}d_{y,1}\\ d_{y,2}\\ d_{y,3}\end{bmatrix}$$

第 6 章

6.5：块$\mathcal{B}_m$ 的运动矢量 $\mathbf{d}m$ 可以由公式(6.2.7)求得，这里应该在$\mathcal{B}_m$ 上求和。这个解只有

当真实的运动矢量很小,或迭代地运用这个公式而且初始估计接近真实的运动矢量时才是成立的。

6.6:

$$\mathbf{a}=\left(\sum_{\mathbf{x}\in\Lambda}[\mathbf{A}(\mathbf{x})]^T\nabla\psi_1(\mathbf{x})\nabla\psi_1(\mathbf{x})^T[\mathbf{A}(\mathbf{x})]\right)^{-1}$$
$$\times\left(\sum_{\mathbf{x}\in\Lambda}(\psi_1(\mathbf{x})-\psi_2(\mathbf{x}))[\mathbf{A}(\mathbf{x})]^T\nabla\psi_1(\mathbf{x})\right)$$

6.8:对于一幅 $M\times M$ 的图像,总的运算次数是$(2R+1)^2M^2$,与块的大小无关。

6.9:$(M/2^{L-1})^2((2R/2^{L-1})+1)^2+18M^2(1-(1/2)^{L-1})$

6.10:对于 EBMA,$720\times480\times65^2=1.46E+9$,对于搜索范围顶层为 16、底层为 1 的两级 HBMA,$360\times240\times33^2+720\times480\times3^2=9.72E+7$。

第8章

8.1:(a)$\log_2 L$;(b)0。

8.6:(a) $H_1=1.5850, H_2=3.0850, H_{c,1}=1.5, \overline{H}=1.5$。(b)$R_1=1.6667, R_2=3.1667, R_{c,1}=1.5$。(c) $R_{\min}=\overline{H}$,由一阶条件哈夫曼编码实现。

8.11:(a)$b=1/\lambda+a$;(b)$a=b/2$;(c)$a=1/\lambda, b=2/\lambda$, $\mathrm{MSE}=(2/\lambda^2)\left(1-\frac{2}{e}\right)$。

8.12:(b) $b_l=(g_l+g_{l+1})/2$。

8.14:MSE = 9/16 每样点。

第9章

9.4:$\sigma_{l,k}^2=\{1+3\rho,1-\rho,1-\rho,1-\rho\}\sigma_s^2=\{3.85,0.05,0.05,0.05\}$, $\sigma_s^2=0.1481$, $G_{TC}=6.75$,其中 $R=2, D_{TC}=0.0093\epsilon_G^2$, $R_k=\{4.35,1.22,1.22,1.22\}$。

9.5:特征值为:$\lambda_k=\{1+3\rho,1-\rho,1-\rho,1-\rho\}\sigma_s^2=\{3.85,0.05,0.05,0.05\}$。对于这种情况,KLT 产生与 DCT 相同的变换系数方差。

9.6:$a_k=\rho/(2\rho+1), k=1,2,3$, $\sigma_p^2=(1-3\rho^2/(2\rho+1))\sigma_s^2=0.06638$, $G_{DPCM}=15.06$,其中 $R=2, D_{DPCM}=0.0041\epsilon_G^2$

9.7:$a_k=\rho/(\rho+1), k=1,2$, $\sigma_p^2=\sigma_s^2(1+\rho-2\rho^2)/(1+\rho)=0.0744$, $G_{DPCM}=13.45$

第10章

10.5:用"0","1","2","-1"代表四个方向,从方向"东"开始,以顺时针顺序,四邻域的直接链码是:

-1,2,-1,2,2,2,0,-1,-1,-1,0,1,1,0,0,-1,-1,-1,1,1,0,1,0,1,1,2,2,1。

差分码是:

-1,-1,1,-1,0,0,2,-1,0,0,1,1,0,-1,0,-1,0,0,2,0,-1,1,-1,1,0,1,0,-1。

10.6:码树是:"0111101101111111"。形状树是:"22132233131",其中,1 代表最亮阴影区,2 代表中等阴影区,3 代表最暗阴影区。

［Ｇｅｎｅｒａｌ Ｉｎｆｏｒｍａｔｉｏｎ］
书名＝视频处理与通信
作者＝
页数＝４４５
ＳＳ号＝１１０９８８２２
出版日期＝